

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

ADA087668

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
	AD-A087 668 (9)	
4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED	
COMPARISON OF OPERATOR AIDED OPTIMIZATION WITH ITERATIVE MANUAL OPTIMIZATION IN A SIMULATED TACTICAL DECISION AIDING TASK,	TECHNICAL REPORT, 3	
6. AUTHOR(s)	7. PERFORMING ORG. REPORT NUMBER	
D. SCHECHTERMAN H. WALSH	TSC-215-67	
8. AUTHORING AGENCY NAME & ADDRESS	9. CONTRACT OR GRANT NUMBER(s)	
INTEGRATED SCIENCES CORPORATION 1640 Fifth Street, Suite 204 Santa Monica, California 90401	N00014-75-C-0811	
11. CONTROLLING OFFICE NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
ENGINEERING PSYCHOLOGY PROGRAM, CODE 455 OFFICE OF NAVAL RESEARCH ARLINGTON, VIRGINIA 22217		
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE	
12-144	11 July 1980	
	13. NUMBER OF PAGES	
	15. SECURITY CLASS. (of this report)	
	UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
16. DISTRIBUTION STATEMENT (of this Report)		
UNLIMITED		
This document has been approved for public release and sale; its distribution is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
Reproduction in whole or in part is permitted for any purpose of the United States Government.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Decision Aids Command and Control Naval Tactics Mission Planning Interactive Graphics Man-Computer Functional Allocation Nonlinear Programming Optimization		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
This is a report of work accomplished on one Task in the project entitled Operational Decision Aids, which was initiated in 1974 by the Office of Naval Research to develop aids for Navy command and control functions and make them available for incorporation in the design of future systems. The report describes the implementation and experimental evaluation of two methods of aiding the selection of (a) an air strike path through a field of ten enemy sensors, and (b) aircraft speeds on each leg of the path. Utility of		

DD FORM 1 JAN 73 1473

404567

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. Abstract (continued)

each candidate strike path was computed according to a predetermined utility function; the utility function was nonlinear and multi-modal.

One of the ISC-designed aids uses Operator Aided Optimization (OAO). An operator using the OAO aid guides a nonlinear programming algorithm by selecting starting points for each solution search by the algorithm. The other aid is called Iterative Manual Optimization (IMO). An operator using IMO selects a candidate solution and the computer acts as a calculator by calculating and displaying the solution's utility. The operator then modifies the previous solution in light of what was learned from seeing its utility. The process is repeated until the operator is satisfied with the utility achieved.

Sixteen college students majoring in technical subjects solved problems with and without the aids and these data were compared. The principal findings from the experiments were:

1. Operators using the aids did significantly better than without them. The average improvement across all subjects and trials for the OAO aid was 9%; the improvement for the IMO aid was 11%. Performance was significantly different across operators for both aids, but this was solely for unaided operation. Thus, the aids served as an "equalizer." They enabled operators having relatively low scores without the aid to do as well as those who had relatively high scores without the aid.
2. Operator performance using OAO was significantly better in a statistical sense than performance using IMO. However, on a percentage basis the advantage of OAO was very slight: an average of 1.3 utility points, or 1.3% (from 98.13 to 99.43 on a scale of 100).

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	
Justification	
By	
Distribution/	
Availability Notes	
Dist	Available or special

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

COMPARISON OF OPERATOR AIDED OPTIMIZATION
WITH ITERATIVE MANUAL OPTIMIZATION IN
A SIMULATED TACTICAL DECISION AIDING TASK

Report No. 215-6

Contract No. N00014-75-C-0811

Prepared for:

Code 455
Director, Engineering Psychology Program
Psychological Sciences Division
Office of Naval Research
Department of the Navy
Arlington, Virginia 22217

By:

Michael D. Schechterman
David H. Walsh

Integrated Sciences Corporation
1640 Fifth Street
Santa Monica, California 90401

July 1980

This document has been approved
for public release and sale; its
distribution is unlimited.

ACKNOWLEDGMENTS

We would like to acknowledge the support of Dr. Martin A. Tolcott, Office of Naval Research. We especially appreciate the guidance and thoughtful suggestions by Dr. Tolcott and Mr. J.R. Simpson of ONR. Their suggestions helped to improve the concepts tested and to increase the information in the analysis.

EXECUTIVE BRIEF

BACKGROUND

This is the fifth report by Integrated Sciences Corporation (ISC) as one of a group of contractors working on the Operational Decision Aids (ODA) program directed by the Office of Naval Research. The ODA program was initiated in 1974. It is intended to develop a variety of decision aids and test and evaluate their usefulness to the Navy. Although the program is not tied to any specific command and control hardware system, it has focused on the functions of a Task Force Commander (TFC) and his staff. The role of ISC has been to find ways to improve man-machine communication by allocating functions between man and machine that take advantage of their respective strengths.

ISC had established in a previous study (Reference 1) that humans are adept at perceiving and sketching complex functional relationships when data that could be used to estimate the function were presented to the human in geometric/graphical format. The following question then arose: How useful is this human capability to perceive such complex relationships? Therefore, ISC proceeded to define (a) two decision aids that used the human capability to solve an experimental program that could also be solved by a fully automated algorithm and (b) an experiment that compared decision performance with and without the aids (Reference 2).

The two ISC-designed aids were called Operator Aided Optimization (OAO) using Nonlinear Programming (NP) and Operator Aided Optimization using Dynamic Programming (DP). Operator performance using the NP aid proved superior to performance with the DP aid and the NP aid was easier for operators to use. In each case the operator controls the use of an algorithm. For the NP aid the operator controls the nonlinear programming algorithm by:

1. Choosing a starting point for the algorithm.
2. Stopping the optimization process of the algorithm when the utility obtained shows diminishing returns versus time.
3. Selecting a new starting point in another region of the solution space.

It is important to understand the purpose of the experiments that were conducted and certain distinctions between the experimental programs and the corresponding real-world situation. The previous experiments with the NP aid were principally designed to contrast decision performance:

1. With the OAO aid versus without the aid.
2. With the OAO aid versus fully automated use of the algorithm.

Although ISC used much of the structure and characteristics of a real-world situation, the experiment was deliberately limited and therefore, in a sense, artificial. The problem situation used in the experiment is the selection of (a) an air strike path through a field of ten enemy sensors and (b) aircraft speeds on each leg of the path. (Hereafter in this report, the selection of path and speeds is abbreviated to "selection of path.") Many aspects of real-world air strike planning were not included in the experimental problem, e.g., aircraft altitude, specific locations of enemy weapon systems, and such real-world systems as electronic countermeasures. Also, the design of the experimental problems made certain perfect-information assumptions in order to simplify the analysis.

The principal findings of the previous experiment with the OAO NP aid were:

1. The operators using the NP aid did significantly better than without the aid. The average improvement across all subjects and trials was 29% with a range of 9% to 123%. Performance was significantly different across operators but this was solely for unaided operation. Thus, the aid served as an "equalizer." It enabled operators having relatively low scores without the aid to do as well as those who had relatively high scores without the aid.

2. The lack of a technical education was apparently not an impediment to good performance with or without either aid.

3. Operator aided optimization was significantly better than automated use of the NP algorithm for both types of rules used by the algorithm to select starting points.

4. A potential implication of the findings is that OAO is attractive to use when it is applicable because:

- a. The operator can see what is happening during the optimization. With pictorial problem representation, he can make adjustments to the optimization procedure or results to compensate for limitations in problem representation more easily than he can when there is no pictorial representation.
- b. The time required to train operators to use OAO with pictorial problem representation is apparently relatively short and does not require technical knowledge of the optimization algorithms.

THE CURRENT STUDY

The study summarized in the previous subsection established that operator performance using the OAO aid on the experimental problems was superior to unaided solution of those problems and to fully automated use of the optimization algorithms. Another mode of aiding the operator, namely, iterative manual optimization (IMO), was suggested as an alternative to OAO after completion of the previous experiment. With IMO an operator would input a solution to the computer and the computer would calculate and display the solution's utility. The operator would then revise the earlier solution based on what was learned from seeing its utility. Thus, with IMO the computer acts as a calculator and all optimization is done by the operator.

The following question then arose: How closely would operator performance using IMO on the experimental problems match performance with OAO? The answer to this question would shed some light on the question of the importance of an optimizing algorithm as part of an aid to solving a complex problem having a multimodal, unsymmetric, nonlinear criterion function. Consequently, ISC designed and performed an experiment to compare IMO with OAO for the same set of problems as were used in the previous experiments.

The analysis of "unaided" operator mode versus performance with the IMO and OAO aids showed similar results. In each case, there was about a 10% average improvement across all operators and trials between the unaided and aided modes. The analysis of IMO versus OAO showed a very slight advantage to OAO: an average of 1.3 points, or 1.3% (from 98.13 to 99.43 on a scale of 100). A questionnaire administered to operators after they completed the experiment showed that they all preferred using the OAO aid rather than the IMO aid. The principal reason given was that the OAO aid was less tedious to use.

The software costs associated with implementing the NP algorithm in the OAO aid were very small because a standardized, already-programmed algorithm was used. One cost of using the NP algorithm is also small because the computer memory space required to store the algorithm only increases the length of the IMO program by 5%.

Another cost difference between IMO and OAO is the amount of CPU use for each. OAO uses the CPU continuously between the time the operator activates "Evaluate" and, later, "Halt." With IMO, the CPU is only used to calculate a few values used in the utility function and the value of the utility function itself once for each set of waypoint and/or speed changes input by the operator. The importance of this difference depends on the amount of OAO use on a computer system that would have many other jobs to run.

The authors believe that the tradeoff between the extra software costs associated with OAO and the performance and ease of use benefits of OAO would favor its selection over the IMO aid.

RECOMMENDATIONS

The real world of air strike planning is more complex than the problem used to compare the OAO and IMO concepts. Important real-world considerations are listed below:

Problem Factors

- capabilities and location of enemy sensors
- capabilities and locations of enemy missiles
- capabilities of enemy fighters and locations of their bases
- capabilities and locations of enemy anti-aircraft guns
- change in location of enemy defense forces during the flight time of the air strike.

Decision Dimensions

- x,y locations of path legs
- speed on each leg
- altitude on each leg
- when to use on-board jamming equipment

Utility Dimensions

- expected damage to strike aircraft due to enemy defenses
- distance between target and strike aircraft when cumulative detection probability exceeds x%
- probability of strike mission success
- fuel remaining

The IMO and OAO aids described in this report account for only a few of the dimensions. Consequently, we recommend the following question for further study:

Is the small magnitude of the performance difference between IMO and OAO observed in the current study due to the relative simplicity of the problems to be solved?

This question may be put another way:

How will the difference in operator performance using IMO- and OAO-type aids change as the dimensions of the problem, decision, and utility function increase?

Performance data from aids and experiments designed to answer this question could be important evidence supporting a decision by R&D program managers to concentrate future efforts on developing one of the aiding concepts in preference to the other.

TABLE OF CONTENTS

EXECUTIVE BRIEF	iii
1.0 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 THE CURRENT STUDY	4
2.0 CONTEXT FOR THE EXPERIMENT	6
2.1 AIR STRIKE SCENARIO	6
2.2 DESCRIPTION OF CONCEPTS TESTED	8
2.3 SIMULATION MODELS AND ALGORITHMS	10
3.0 SYSTEM OPERATION	20
3.1 ITERATIVE MANUAL OPTIMIZATION (IMO)	20
3.2 OPERATOR AIDED OPTIMIZATION (OAO)	24
4.0 DESCRIPTION OF THE EXPERIMENTS	30
4.1 EXPERIMENTAL DESIGN	30
4.2 OPERATORS AND TRAINING OF OPERATORS	34
4.3 EXPERIMENTAL PROCEDURE	35
5.0 RESULTS	36
5.1 ANALYSIS OF UNAIDED OPERATOR AGAINST IMC DATA	36
5.2 ANALYSIS OF UNAIDED OPERATOR AGAINST OAO DATA	39
5.3 ANALYSIS OF IMO AGAINST OAO	41
5.4 FOLLOW-UP QUESTIONNAIRE	43
6.0 DISCUSSION OF RESULTS	47
6.1 SUMMARY OF EXPERIMENTAL FINDINGS	47
6.2 PERFORMANCE DIFFERENCES IN THE UNAIDED MODE	47
6.3 SOFTWARE COSTS ASSOCIATED WITH IMPLEMENTING AND USING THE OPTIMIZING ALGORITHM	49
6.4 RECOMMENDATIONS	51
REFERENCES	53
APPENDIX A: NONLINEAR PROGRAMMING OPTIMAL PATH SOLUTION	54
APPENDIX B: TRAINING MATERIALS FOR IMO	59
APPENDIX C: TRAINING MATERIALS FOR OAO	93
DISTRIBUTION LIST	129

LIST OF FIGURES

Figure		Page
1	Strike Scenario Geography	7
2	Interrelation of Models and Algorithms	11
3	Single-Sensor Detection Rate as a Function of Range	13
4	Performance Curve of Modeled Sensor	14
5	Display Appearance at Beginning of Problem to Be Solved	21
6	Function Button Configuration	22
7	Display Appearance After the Operator Has Input His First Starting Point	25
8	Display Appearances After the Operator Has Finished Exploring the Region Around the Starting Point on Figure 7.	26
9	Typical Plot of Utility Versus Function Evaluations	28
10	Interaction Between Prior Experience and Concepts in the IMO Experimental Results	38
11	Interaction Between Prior Experience and Concepts in the OAO Experimental Results	38
12	Interaction Between Replications and Concepts in the OAO Experimental Results	40
13	Interaction Between Prior Experience and Concepts in the IMO vs. OAO Experimental Results	40
14	Best-to-Date Normalized Utility Versus Time for OAO and IMO	44
15	Time Averaged Scoring Rule Utility Versus Time for OAO and IMO	45

LIST OF TABLES

Table		Page
1	Fuel Consumption Rates	17
2	Unaided Operator vs. IMO ANOVA (with pooling)	37
3	Unaided Operator vs. OAO ANOVA (with pooling)	37
4	IMO vs. OAO ANOVA (with pooling)	42

1.0 INTRODUCTION

1.1 BACKGROUND

This is the fifth report by Integrated Sciences Corporation (ISC) as one of a group of contractors working on the Operational Decision Aids (ODA) program directed by the Office of Naval Research. The ODA program was initiated in 1974. It is intended to develop a variety of decision aids and test and evaluate their usefulness to the Navy. Although the program is not tied to any specific command and control hardware system, it has focused on the functions of a Task Force Commander (TFC) and his staff. The role of ISC has been to find ways to improve man-machine communication by allocating functions between man and machine that take advantage of their respective strengths.

ISC had established in a previous study (Reference 1) that humans are adept at perceiving and sketching complex functional relationships when data that could be used to estimate the function were presented to the human in geometric/graphical format. The following question then arose: How useful is this human capability to perceive such complex relationships? Therefore, ISC proceeded to define (a) two decision aids that used the human capability to solve an experimental problem that could also be solved by a fully automated algorithm and (b) an experiment that compared decision performance with and without the aids (Reference 2).

The two ISC-designed aids were called Operator Aided Optimization (OAO) using Nonlinear Programming (NP) and Operator Aided Optimization using Dynamic Programming (DP). In each case the operator controls the use of an algorithm. For the NP aid the operator controls the algorithm by:

1. Choosing a starting point for the algorithm.
2. Stopping the optimization process of the algorithm when the utility obtained shows diminishing returns versus time.
3. Selecting a new starting point in another region of the solution space.

The operator controls the DP aid by constraining the range of values for each variable considered by the algorithm.

It is important to understand the purpose of the experiments that were conducted and certain distinctions between the experimental problems and the corresponding real-world situation. The previous experiments were principally designed to contrast decision performance:

1. With an OAO aid (NP or DP) versus without the aid.
2. With an OAO aid versus fully automated use of the algorithm (NP or DP).

Although ISC used much of the structure and characteristics of a real-world situation, the experiment was deliberately limited and therefore, in a sense, artificial. The problem situation used in the experiment is the selection of (a) an air strike path through a field of ten enemy sensors and (b) aircraft speeds on each leg of the path. (Hereafter in this report, the selection of path and speeds is abbreviated to "selection of path.") Many aspects of real-world air strike planning were not included in the experimental problem, e.g., aircraft altitude, specific locations of enemy weapon systems and such real-world systems as electronic countermeasures. Also, the design of the experimental problems made certain perfect-information assumptions in order to simplify the analysis.

The principal findings of the previous experiment were:

1. The operators using the NP aid did significantly better than without the aid. The average improvement across all subjects and trials was 29% with a range of 9% to 123%. Performance was significantly different across operators but this was solely for unaided operation. Thus, the aid served as a "equalizer." It enabled operators having relatively low scores without the aid to do as well as those who had relatively high scores without the aid.

2. Operators using the DP aid did significantly better than without the aid. The average improvement across all subjects and trials was 12% with a range of 3.5% to 27%.

3. The lack of a technical education was apparently not an impediment to good performance with or without either aid.

4. Operator aided optimization was significantly better than automated use of the NP algorithm for both types of rules used by the algorithm to select starting points.

5. The NP aid was less complex to use than the DP aid and operators generally preferred working with the NP aid to working with the DP aid. Operators using OAO with the NP aid found the global optimum on a higher percentage of trials than operators using OAO with the DP aid. The average time required to adequately train an operator to use either aid was about four hours.

6. A potential implication of the findings is that OAO is attractive to use when it is applicable because:

- a. The operator can see what is happening during the optimization. With pictorial problem representation, he can make adjustments to the optimization procedure or results to compensate for limitations in problem representation more easily than he can when there is no pictorial representation.
- b. The time required to train operators to use OAO with pictorial problem representation is apparently relatively short and does not require technical knowledge of the optimization algorithms.

1.2 THE CURRENT STUDY

The study summarized in the previous subsection established that operator performance using the OAO aid on the experimental problems was superior to unaided solution of those problems and to fully automated use of the optimization algorithms. Another mode of aiding the operator, namely, iterative manual optimization (IMO), was suggested as an alternative to OAO after completion of the previous experiment. With IMO an operator would input a solution to the computer and the computer would calculate and display the solution's utility. The operator would then revise the earlier solution based on what was learned from seeing its utility. Thus, with IMO the computer acts as a calculator and all optimization is done by the operator.

The following question then arose: How closely would operator performance using IMO on the experimental problems match performance with OAO? The answer to this question would shed some light on the question of the importance of an optimizing algorithm as part of an aid to solving a complex problem having a multimodal, unsymmetric, nonlinear criterion function. Consequently, ISC designed and performed an experiment to compare IMO with OAO for the same set of problems as were used in the previous experiments.

All phases of this study are documented in the following sections. Section 2 describes the way the basic path optimization problem was constructed. It explains how the ONRODA Scenario (Reference 3) was adapted, distinguishes at more length between the OAO and IMO modes of determining strike path solutions, explains the analytical models for single sensor detection performance and aircraft fuel consumption, and characterizes the utility function developed to evaluate strike paths. Section 3 details system operation; it comprises step-by-step explanations of how path

solutions were obtained by subjects using the OAO and IMO aids. Sections 4 and 5 outline the experiment and the data analyses performed, respectively. Section 6 interprets the results insofar as the data warrant. The appendices document the NP algorithm used in the OAO aid and the training materials provided to operators.

2.0 CONTEXT FOR THE EXPERIMENT

A tactical decision task was defined to investigate the usefulness of decision aids that make use of man's ability to visually perceive complex functional relationships. The task was that of optimizing an air strike path through a defender's multi-sensor detection field. This section describes the task scenario, the system concepts that represent different ways of optimizing a strike path, and the models of the scenario variables that constitute the experimental vehicle. Each model described reflects certain assumptions made about the behavior of the scenario variable. These assumptions, in turn, were adopted to keep the test vehicle simple, rather than to faithfully model the variables' "real world" performance. The utility criterion function, ultimately used as a performance measure, is also described here in terms of its supporting models.

2.1 AIR STRIKE SCENARIO

The problem selected, implicit in the ONRODA scenario, was that of optimizing an air strike path between a strike launch point and a target. The evaluation of the path depended on the probability of an aircraft's being detected by the enemy and the amount of fuel consumed by the aircraft along the strike path. Accordingly, certain elements of interest, particularly the scenario geography, were extracted from the ONRODA Warfare Scenario (Reference 3), and other details, described below, were added. The scenario developed here assumes that the decision has been made to conduct an air strike against ONRODA, so that investigating the relative usefulness of competing decision aids in this study means applying them to one aspect of the operational implementation of the decision to strike.

Figure 1 shows the 500-by-500 n.m. portion of the ONRODA warfare scenario area map used to provide the geographical context for this study. The boundaries provide an area west of ONRODA for the selection of strike launch points and (it is assumed) enough room to plan strike paths that do not violate the ORANGE sanctuary.

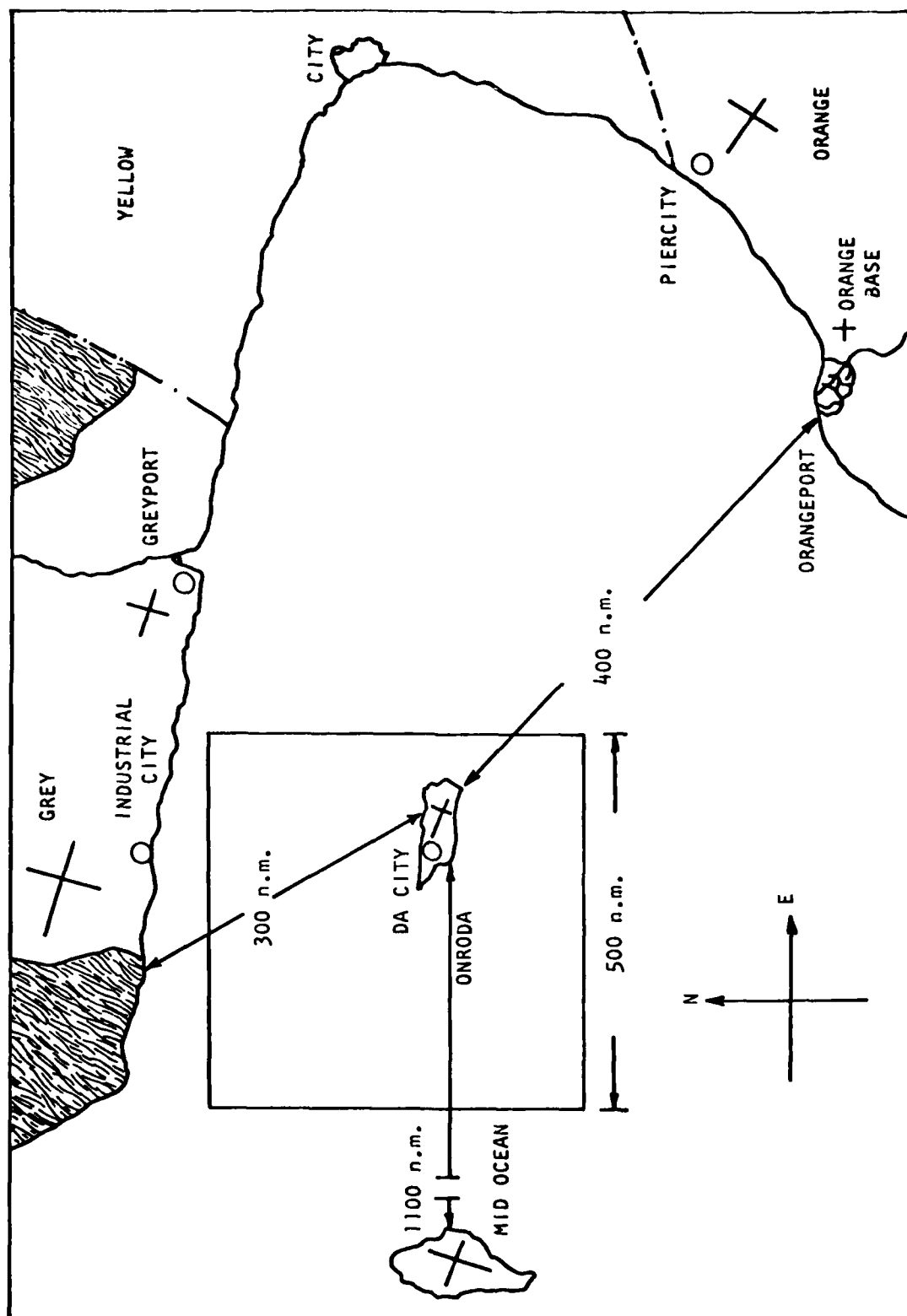


Figure 1. Strike Scenario Geography

The air strike scenario used here incorporates some further assumptions. First, the strike target is taken to be the ONRODA airfield complex only. Second, the strike aircraft are supersonic, and they carry suitable stores and a predetermined fuel allotment.

The assumptions made about enemy defense have to do with the number, locations, and ranges of the ORANGE sensors that are capable of detecting the strike aircraft. Own intelligence reports that there are ten such sensors and that their locations are pinpointed. One sensor is installed on ONRODA near the airfield. The other nine are ocean-platform mounted, and since ORANGE knows the general location of the task force, they are positioned west of ONRODA between the island and the task force. Intelligence reports that all the ORANGE sensors are the same type and have the same detection performance capability. The problem is to plan a strike path against the airfield on ONRODA that (a) minimizes the probability of strike aircraft being detected, given the locations and types of enemy sensors, and (b) does not impose excessive fuel requirements on the aircraft, given the fuel allotment and the fuel consumption characteristics.

Further assumptions for this scenario are that neither the enemy's defense nor airborne enemy aircraft are to be considered explicitly as strike factors. Implicitly, enemy defense capability is one reason to minimize the probability of being detected along the strike path, tantamount to considering surprise as a strike factor. In a similar manner, attempting to postpone detection also affords less time for ORANGE aircraft on ONRODA or on the mainland to react, while attempting to conserve fuel enables the strike aircraft to maneuver if challenged by ORANGE aircraft after reaching the strike target.

2.2 DESCRIPTION OF CONCEPTS TESTED

Solving the strike path selection problem requires choosing path way points between the start point and target and specifying aircraft speeds along each leg. Two procedures for solution of the best path problem were chosen for study. One is the operator-aided optimization (OAO) method

using nonlinear programming, as initially described in Reference 2 and found to be the best procedure studied there. The second procedure is an iterative manual optimization (IMO) method described below. The two procedures were compared with each other, and with an "operator-unaided" mode in which the operator specified a single set of path way points and speeds on path legs which constitute his estimate of the best solution. The latter is a one-step process; optimization is not done progressively over time as occurs in OAO and IMO. The operator-unaided mode corresponds to the procedure that would be used today by a Task Force Commander, with the exception that the operator in this mode has the use of the same contours representing composite detection capability of enemy sensors that are available to the other methods. (Thus, the operator-unaided mode is not a completely unaided mode.)

The IMO procedure is essentially an iterative version of the operator unaided mode. Each trial solution involves the choice of the way point positions and the speeds along each leg. The operator has pictorial information about the scenario and alphanumeric information such as the utility of the current trial solution and of the best trial solution to date. Using this information, the operator attempts to converge manually on the best path.

In the OAO method, the operator enters a trial solution as a "starting point" and then starts the nonlinear programming optimization algorithm. The algorithm then iteratively improves on the solution while displaying the results in a format nearly identical to IMO. All optimization is done locally, i.e., the algorithm cannot "see" better paths which are hidden behind "hills" in the solution space (this is typical of nonlinear programming optimizers). The operator decides when to stop the algorithm and select a new starting "point" trial solution.

2.3 SIMULATION MODELS AND ALGORITHMS

The "goodness" of a path generated under any of the system concepts depends on two factors: fuel consumed along the path and the cumulative probability of being detected. In order to compute a numerical value (or utility) that reflects a given path's "goodness," it is first necessary to have some way of quantifying those two factors. This was provided by a set of simulation models and computational algorithms. Fuel consumption was modeled as a single functional relationship. The cumulative probability of being detected, however, is more complex and depends on how the characteristics of the detection field are defined. In general, this involves first defining single-sensor performance, then defining the way a number of these single sensors combine to create a composite detection field.

The set of models and algorithms used in the study includes:

1. Single-Sensor Detection Rate Model
2. Cumulative Probability of Being Detected Algorithm
3. Fuel Consumption Model
4. Utility Criterion Function
5. Nonlinear Programming Algorithm
6. True Detection Rate Contour-Drawing Algorithm

Numbers 1-4 are described in this section; number 5 is documented in Appendix A; number 6 is documented in a previous report (Reference 2, Appendix A).

2.3.1 Overview

Figure 2 shows how the models and algorithms are used. Scenario elements (composite detection capability of the ten enemy sensors, strike launch point, and target) defining the problem are stored in the computer and are shown to the operator by means of the display (①). He enters his inputs to the IMO or OAO algorithm by means of the display peripherals (②). Inputs to both algorithms are the problem definition and the operator inputs.

0AO only: -----

IMO only: -----

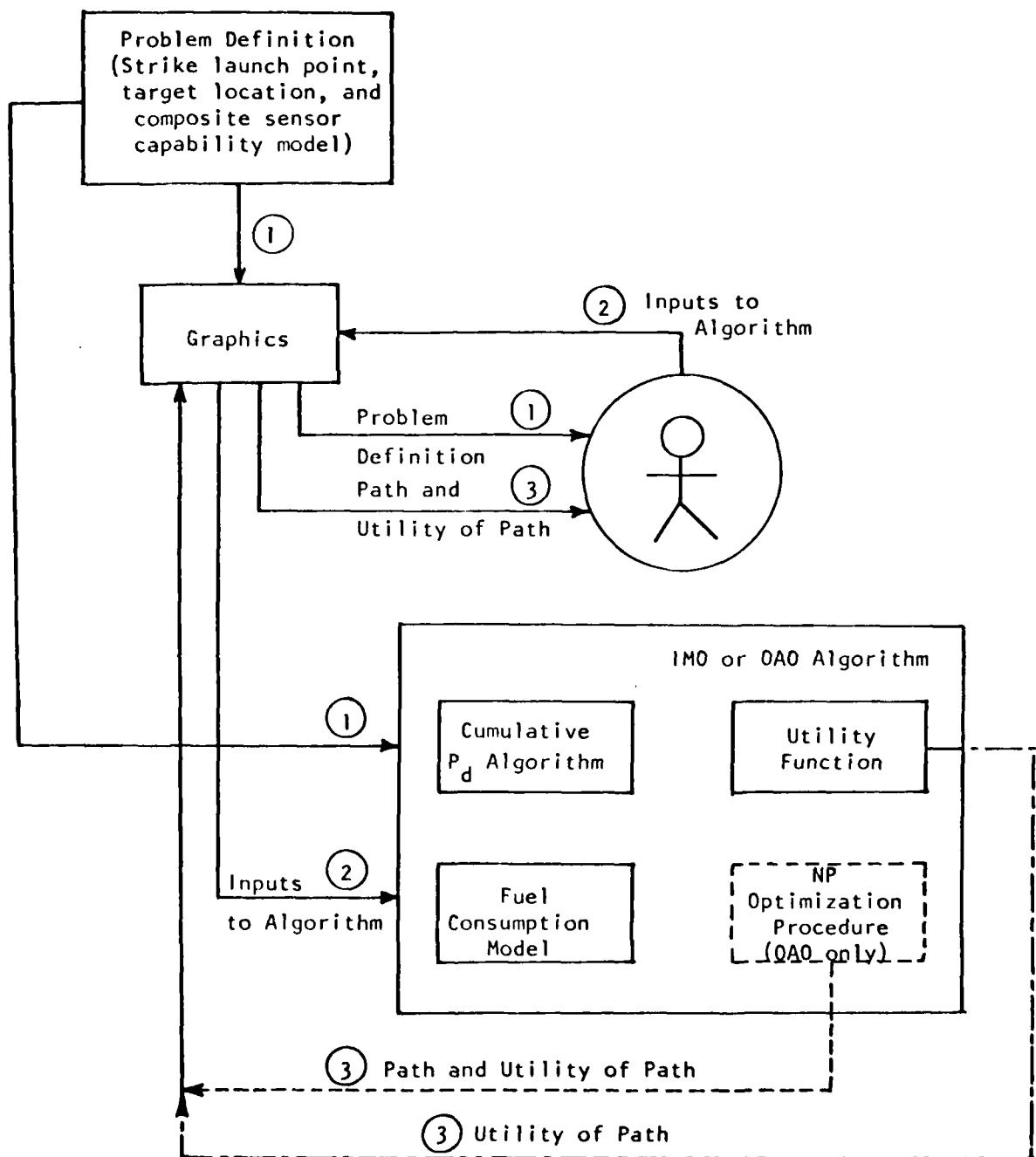


Figure 2. Interrelation of Models and Algorithms.

Each algorithm considers a candidate path, finds the cumulative probability that the air strike will be detected if that path is used, the fuel consumed on the path, and the utility of the path considering both cumulative detection probability and fuel consumption. Each path considered and its utility are displayed to the operator ((3)). In OAO, the NP optimization procedure continues to find better paths, and these are continually displayed until the operator decides to use new inputs. In IMO, no such procedure occurs; the algorithm simply waits for a new candidate path to be input by the operator and the utility function is used to calculate the value of a candidate path when provided.

2.3.2 Single-Sensor Detection Rate Model

The detection capability for a single human-operator sensor is modeled as a detection rate,^{*} which gives the probability of detection per time unit. The detection rate is assumed to vary as a function of range from the sensor. This relationship can be quantified according to:

$$\gamma(R) = \frac{\gamma_{\max} \sqrt{e}}{R_{\max}} R \cdot \exp\left(\frac{-R^2}{2R_{\max}^2}\right) \quad (1)$$

where

$\gamma(R)$ = the value of detection rate at radial distance R from sensor

γ_{\max} = maximum detection rate for the sensor

R_{\max} = range from sensor at which γ_{\max} occurs

The general shape of detection rate-versus-range curve as governed by Equation (1) is shown in Figure 3. Equation (1) for $\gamma(R)$ models a sensor with a maximum detection rate γ_{\max} at range R_{\max} from the sensor. From

*Detection rate is a quantitative measure of sensor performance (Ref. 4) defined over the space surrounding a sensor. An intuitive understanding of detection rate, $\gamma(x,y)$, may be had by considering that $\gamma\Delta t$ is the conditional probability that a target is detected at or near (x,y) given that 1) Δt is small and 2) no detection occurred before Δt .

this range, R_{\max} , the detection rate drops off monotonically moving away from the sensor, approaching zero at some range beyond R_{\max} . Hence, if we visualize $\gamma(R)$ as a three-dimensional surface, it would look like a volcano with a hole at the center, where the sensor is located. Around this hole is a circular ridge at a radial distance R_{\max} from the center of the hole. Beyond the ridge the sides of the "volcano" may slope downwards until "ground level" is reached.

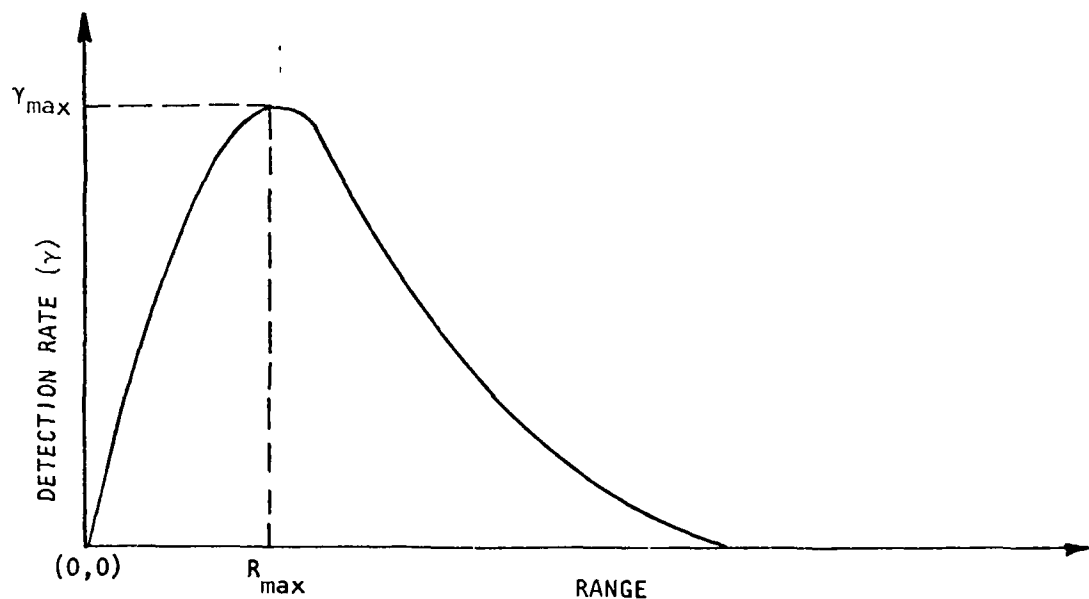


Figure 3. Single-Sensor Detection Rate as a Function of Range.

For the experiment one type of sensor was defined, corresponding to $R_{\max} = 37.5$ nautical miles. This value of R_{\max} was selected for its suitability to the study. It was not intended to be the performance value for any "real world" sensor. The maximum detection rate γ_{\max} was 0.1. The performance curve for the sensor type is shown in Figure 4.

Recall that the scenario specified ten enemy sensors deployed, so that if two (or more) detection ranges overlap, we are really concerned about our strike aircraft being detected by at least one sensor rather than being detected by more than one. In other words, we are concerned about the total detection

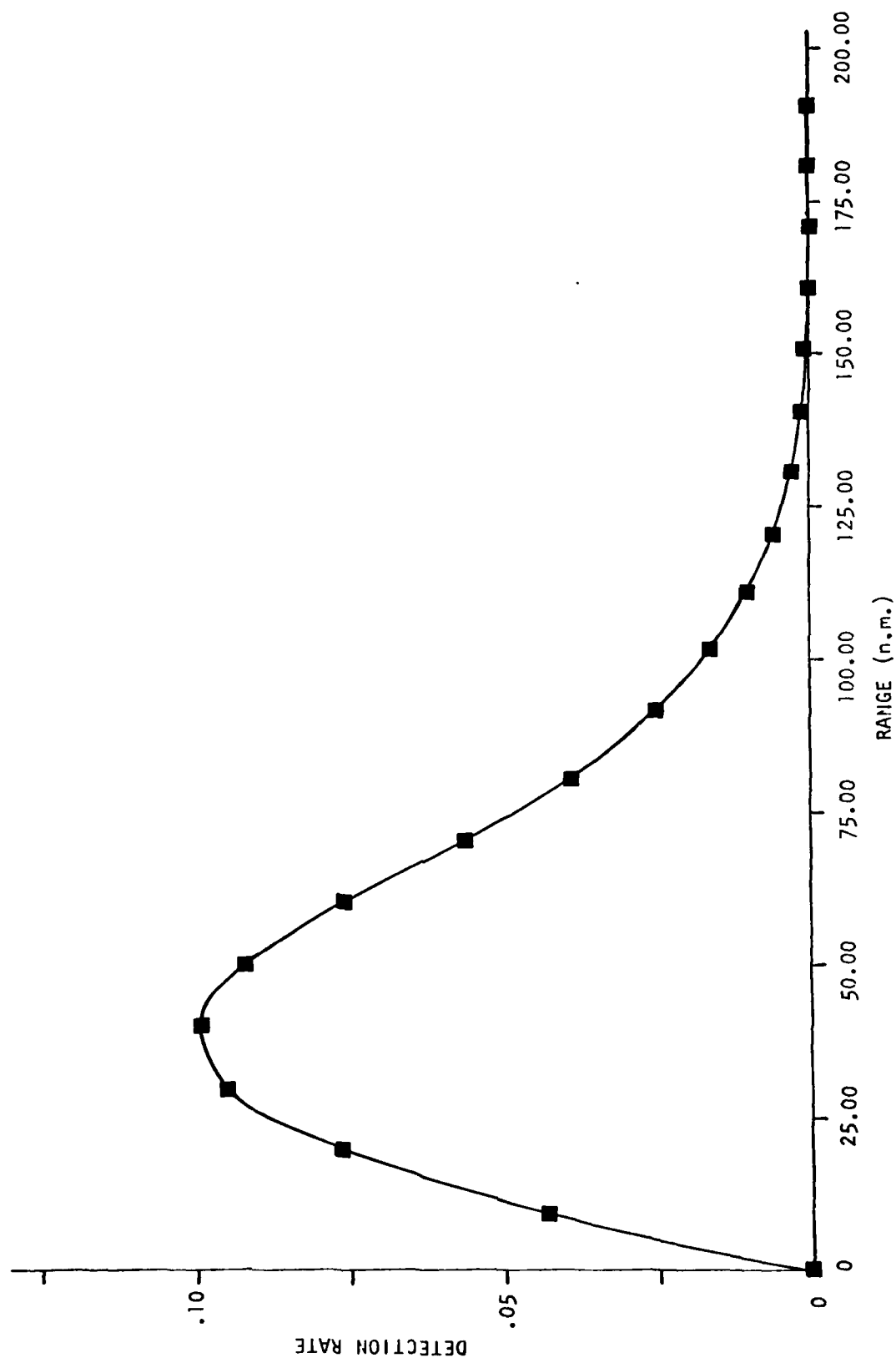


Figure 4. Performance Curve of Modeled Sensor.

rate at any point that any given set of sensor locations will produce. This composite detection rate is easily computed. The composite detection rate γ_c at a point (x,y) is the sum of detection rates at (x,y) due to each sensor. Hence, if $\gamma_i(x,y)$ is the detection rate due to sensor i , the composite detection rate at (x,y) is

$$\gamma_c(x,y) = \sum_i \gamma_i(x,y) \quad (2)$$

Each term $\gamma_i(x,y)$ on the right hand side of Eq.(2) is obtained by transforming the radial coordinates of Eq.(1) into rectangular coordinates.

The reader may question the validity of the additive operation in Eq. (2), since probabilities are not additive in general. After all, detection rate as we have defined it is the probability of detection per unit time. The justification of the operation in Eq. (2) lies in the fact that we choose Δt (see footnote on page 12) small enough such that within Δt the probability of detection by two or more sensors is negligible,* all the higher order terms in the exact expansion for the left hand side of Eq. (2) drop out, leaving the right hand side of Eq. (2).

2.3.3 Cumulative Probability of Being Detected Algorithm

For an aircraft flying an air strike path through the enemy's multi-sensor detection field, it is necessary to calculate the cumulative probability that the aircraft will be detected by the time it reaches the target. The cumulative probability that an aircraft will not be detected on a given leg by a single sensor is the building block used to calculate cumulative detection probability. This is:

*This may remind the reader of similar practices in various branches of operations research, such as queueing theory.

$$P_{nd} \text{ (cumulative, no detection, single sensor)} = \exp \left[- \int_{t_0}^{t_1} \gamma[R(t)] dt \right] \quad (3)$$

where:

t_0 = time at beginning of leg

t_1 = time at end of leg

For multiple sensors, the cumulative probability that an aircraft will not be detected on a given leg is:

$$P_{nd} \text{ (cumulative, no detection on leg)} = \exp \left[- \sum_{s=1}^{s=S} \int_{t_0}^{t_1} \gamma[R_{s,j}(t)] dt \right] \quad (4)$$

where

S = total number of enemy sensors

The cumulative detection probability for the entire path is calculated by:

$$P_d \text{ (cumulative detection on path)} = 1 - \exp \left[- \sum_{\ell=1}^{\ell=L} \sum_{s=1}^{s=S} \int_{t_{0,\ell}}^{t_{1,\ell}} \gamma[R_{s,j}(t)] dt \right] \quad (5)$$

where:

L = number of legs in path

2.3.4 Fuel Consumption Model

The rate of fuel consumption was calculated in accordance with Equation 6 below:

$$\text{Fuel rate} = 0.0377 v^2 - 16.57v + 3869 \text{ (lbs/hr)} \quad (6)$$

where,

v = aircraft speed in knots

Fuel used per path leg is:

$$\text{Fuel consumed (leg}_i\text{)} = \frac{(\text{Leg Length}) (\text{Fuel Rate (v}_j\text{)})}{v_j} \quad (7)$$

Operators were allowed to select any speed from 250 to 1,000 knots for each leg. The fuel consumption rates for three representative speeds are listed in Table 1. The second and third columns of Table 1 are equivalent; they are simply expressed in different units for easier reference.

The amount of fuel that an aircraft carries on each mission is proportional to the range from the strike launch point to the target. Thus, if the range is doubled, the fuel allowance for the mission also doubles. The fuel allowance for each nautical mile between the air strike start position and target was 39.69 pounds. This permits the aircraft to do some high-speed

Table 1. Fuel Consumption Rates

Velocity (knots)	Fuel Consumption Rates	
	lbs/sec	lbs/n.m.
250	0.5785	8.331
625	2.289	13.183
1,000	6.944	24.999

maneuvering, but sustained high-speed travel is discouraged by the fact that allotted fuel would run out before the aircraft could accomplish the mission or return to the carrier.

2.3.5 Utility Criterion Function

A utility criterion function with which to measure the performance under each of the system concepts in the experiment was defined. The problem was to select an optimal air strike path, so an appropriate utility criterion function is one which measures the "goodness" of such an air strike path. The two variables selected to determine the goodness of an air strike path were fuel consumption along the path and probability of being detected by the enemy sensors (Subsections 2.3.3 and 2.3.4, preceding). Since the utility function was predefined to measure the goodness of any proposed path, no inputs were elicited from operators as to desirable values of the two component variables. The following definition of the utility criterion function, U , incorporates a tradeoff between minimizing the probability of being detected by enemy sensors on one hand and maximizing the fuel remaining upon arrival at the target on the other.

$$U(F,P) = 100 \left[\frac{(a - b)D - F}{2(a - 2b)D} \right]^{(.01 + 4.95P)}, \text{ if } (a - b)D - F \geq 0 \quad (8)$$
$$= 100[(a - 2b)D - F], \text{ if } (a - b)D - F < 0$$

where

F = total amount of fuel consumed upon arrival at target

P = cumulative probability of being detected by enemy sensors

D = distance between strike launch point and target

a = fuel allowance/n.m.

b = fuel consumption/n.m. at an achievable speed resulting in the lowest fuel consumption per unit distance traveled

For each mission the fuel allowance is proportional to the shortest distance between the air strike launch point and the target ($a : D$). The absolute minimum fuel that has to be preserved in order to return from the target is ($b \cdot D$). Hence, $(a - b)D$ is the maximum amount of fuel available

for maneuvering to the target, and $(a - 2b)D$ is the maximum amount of fuel remaining upon return to the carrier. Note that if the aircraft runs out of fuel before returning to the carrier, the resulting utility is negative and equal to the difference between minimum possible fuel usage and actual usage for the NP algorithm. This is a device to increase convergence speed. For the experiment, $a = 39.69$ lbs/n.m., and $b = 8.3$ lbs/n.m., corresponding to a velocity of 250 n.m./hr.

The utility function takes on any value between 0 and 100 (except for the negative values noted above), with higher utility values corresponding to "better" paths. As the probability of being detected by enemy sensors decreases, the utility value goes up. Also, if the probability remains constant, the utility value increases as fuel consumption drops. It is obvious why it is desirable to minimize the probability of being detected by enemy sensors. The rationale for encouraging fuel preservation is that if detection occurs at any time up to arrival at the target, there should be as much fuel left as possible for flight maneuvering to try to return safely.

In general, the two goals of minimizing fuel consumption and minimizing the probability of being detected are incompatible. A nontrivial optimal air strike path thus requires a reasonable compromise between the two goals. The utility function was designed as representative of the class of functions useful in selecting an air strike path through a multi-sensor field, and Eq.(8) embodies a trade-off between remaining fuel and cumulative probability of being detected.

3.0 SYSTEM OPERATION

3.1 ITERATIVE MANUAL OPTIMIZATION (IMO)

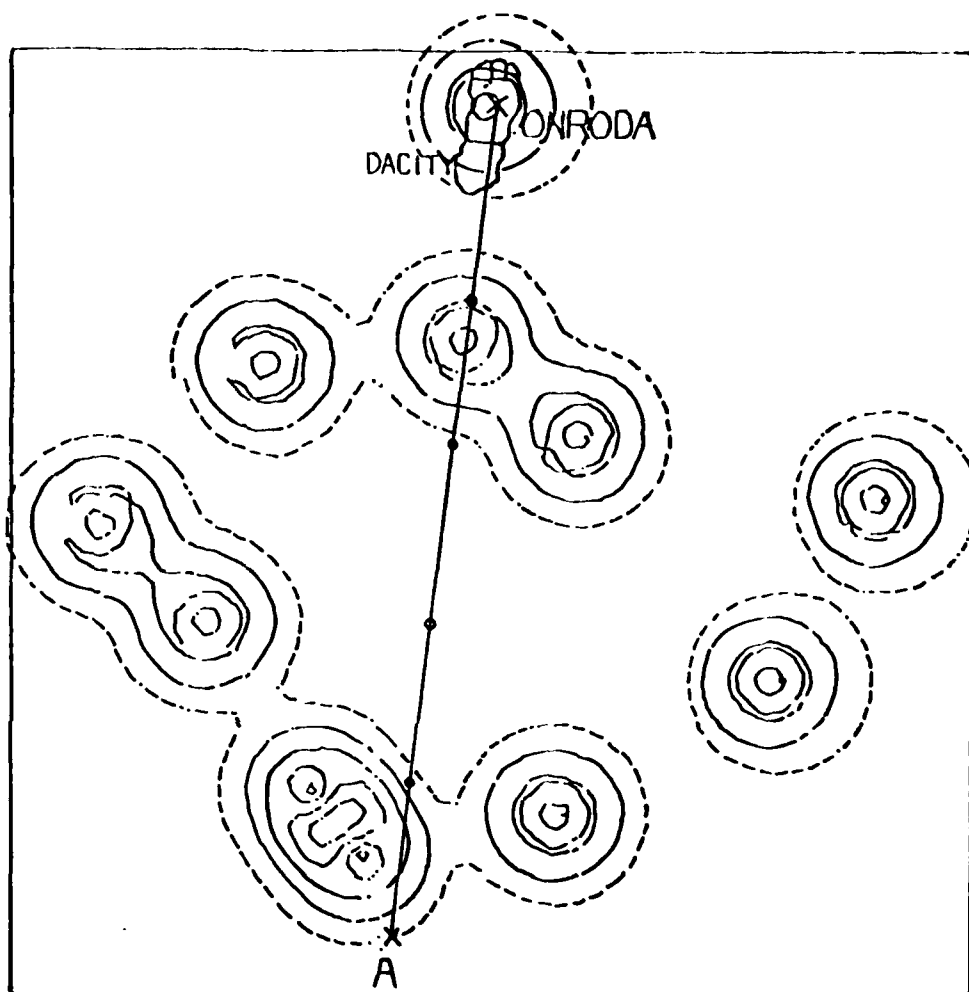
The setup for Iterative Manual Optimization (IMO) includes the starting "point" or first trial solution. In the air strike problem, the starting "point" is (a) five path legs connecting the air strike launch point and the target and (b) speeds for each leg. The legs are specified by picking four "way points" between the launch point and target. Speeds are selected from a range of 250 to 1000 knots. After the start point has been specified, the operator may attempt to find a better combination of way points and speeds. He or she does this by exploring changes in the location of each way point and the speed for each leg, and looking for any improvement in the path utility.

At the beginning of a problem the display appears as shown in Figure 5. The path from launch point to target is a straight line with way points indicated at $1/5$, $2/5$, $3/5$, and $4/5$ of the straight line distance. Speed for each leg is initially set by the program at 600 knots as indicated under "INIT" at the left in Figure 5. The subject uses the appropriate buttons on a function button box (see Figure 6) and a joystick to change the position of the four way points. He uses the appropriate function buttons and number key on an adjacent keyboard to change speed on any leg.

The subject's purpose is to investigate as many reasonable potential solution regions as possible in 15 minutes. As soon as the problem is shown on the display, the subject must decide what region he wants to explore first. He is to pick the region that he thinks is most likely to contain the best solution. He then changes the locations of the way points and speeds accordingly. The resultant path and speeds constitute his estimate of the best solution and correspond to the "unaided operator" concept.

MINS
.1

LEG	INIT	CURR
1	600	600
2	600	600
3	600	600
4	600	600
5	600	600



FUNCTION EVALUATIONS

PROB =
FUEL =
UTILITY =
BEST UTILITY TO DATE =

Figure 5. Display Appearance at Beginning of Problem to be Solved.

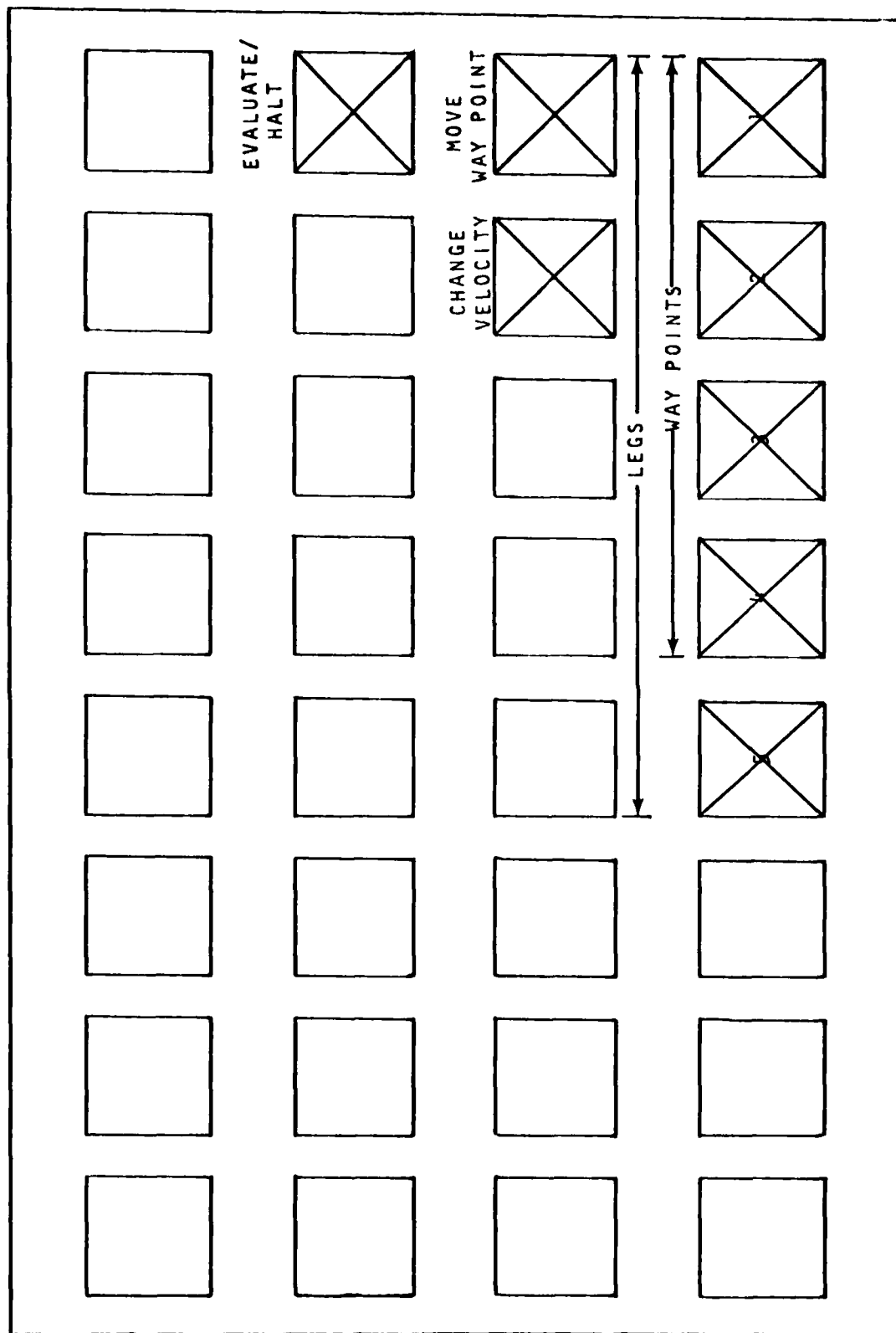


Figure 6. Function Button Configuration

At the beginning of the problem, the three buttons designated as "Evaluate/Halt," "Change Velocity," and "Move Way Point" are lighted on the box. In order to move a way point, the operator pushes that button. When this is done, the four buttons marked 1, 2, 3, and 4 will light. Then the operator pushes the button corresponding to the point to be moved, i.e., 1, 2, 3, or 4. Way point 1 is the closest to the beginning of the strike path and 4 is the nearest to the end (ONRODA Island). Moving the way point is accomplished with the joystick. When a single way point is changed, a second way point can be changed by pressing "Move Way Point" and the appropriate number of the way point. The act of pressing "Move Way Point" records the position of the last way point that was changed.

To change a speed on one of the five legs, the operator pushes "Change Velocity." The five buttons marked 1, 2, 3, 4, and 5 light. Leg 1 refers to the leg closest to the path start point and leg 5 refers to the path closest to the end point (ONRODA Island). Then the operator pushes the button corresponding to the leg for which he wants to change speed and:

1. Uses the teletype keyboard to input the speed he wants used on the selected leg. A decimal point is put at the end of the number. (This is essential.)
2. Pushes the teletype key marked "CR."

Thus, if he wanted to change the speed on leg 3 to 850 knots, he would:

1. Press function button "Change Velocity"
2. Press function button "3"
3. Press teletype key "8"
4. Press teletype key "5"
5. Press teletype key "0"
6. Press teletype key "."
7. Press teletype key "CR"

When the operator has changed all the way points and speeds to those he wants, he then presses the function button marked "Evaluate/Halt." The program will then evaluate the starting point consisting of the four way points and five speeds. The values for probability of detection, fuel consumed, and path utility will be displayed in the lower right corner of the screen (refer back to Figure 5). The number of paths tried is recorded as "function evaluations" in the lower left corner. Figure 7 shows what the display might look like after the operator has input his first starting point. Figure 8 shows what the display might look like after the operator has finished exploring the region around the starting point in Figure 7.

As a guide to optimum use of problem time, the operator should count the number of regions that could reasonably be expected to contain the best path. Dividing 15 minutes by the number of regions to be explored indicates approximately the number of minutes the operator should devote to each region. Depending on the problem, there will be enough time to explore 3, 4, or 5 regions.

At the end of 15 minutes the computer will have stored:

1. The utility of the path comprised of the first way points and leg speeds entered by the operator.
2. The utility of each best-solution-to-date at the end of each minute, excluding the first minute.

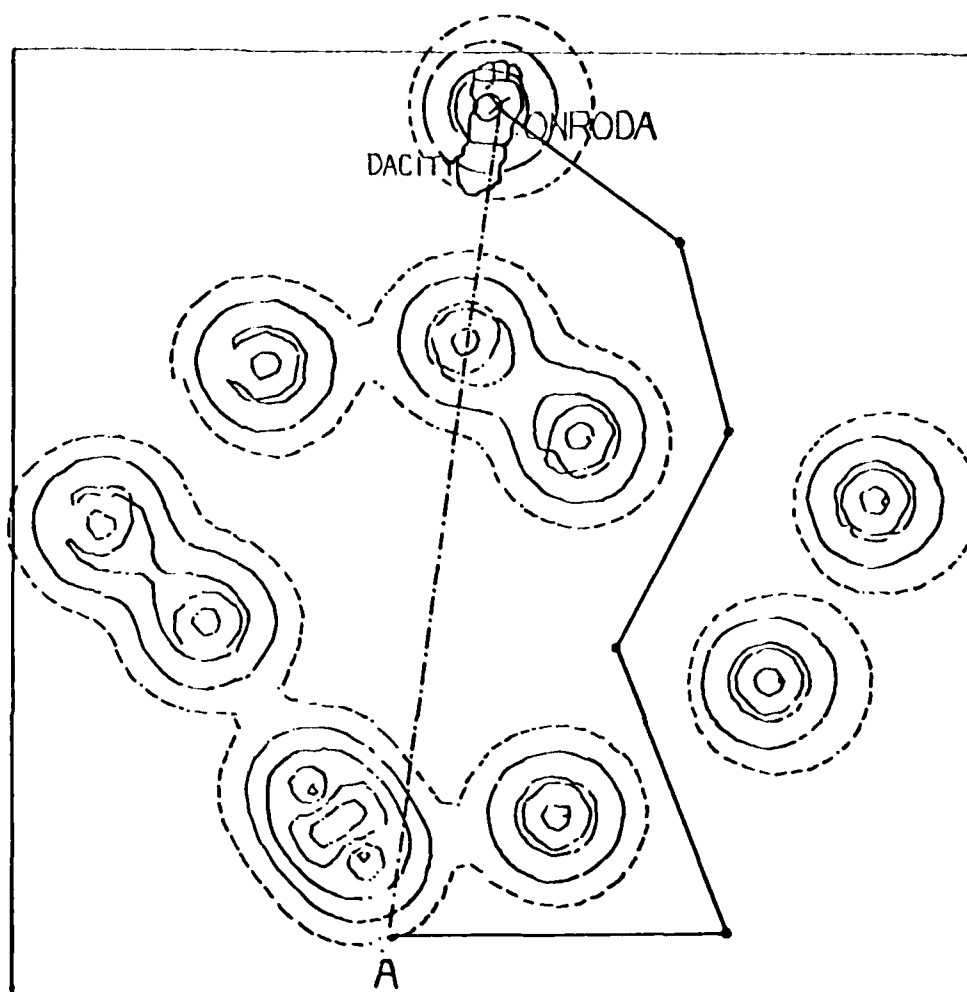
These are the data that are used in the analysis of operator-generated data.

3.2 OPERATOR AIDED OPTIMINATION (OAO)

The setup for the nonlinear programming (NP) technique used in Operator Aided Optimization (OAO) is the same as for IMO, that is, the function button configuration (Figure 6) and the appearance of the display are the same as for IMO. Moving way points and changing speeds for path legs are done in the same way as for IMO. After the start point has been specified, the NP technique operates to find a better combination of way points and speeds. It does this by exploring changes

MINS
1.5

SPEED (KTS)		
LEG	INIT	CURR
1	600	380
2	600	270
3	600	270
4	600	520
5	600	600



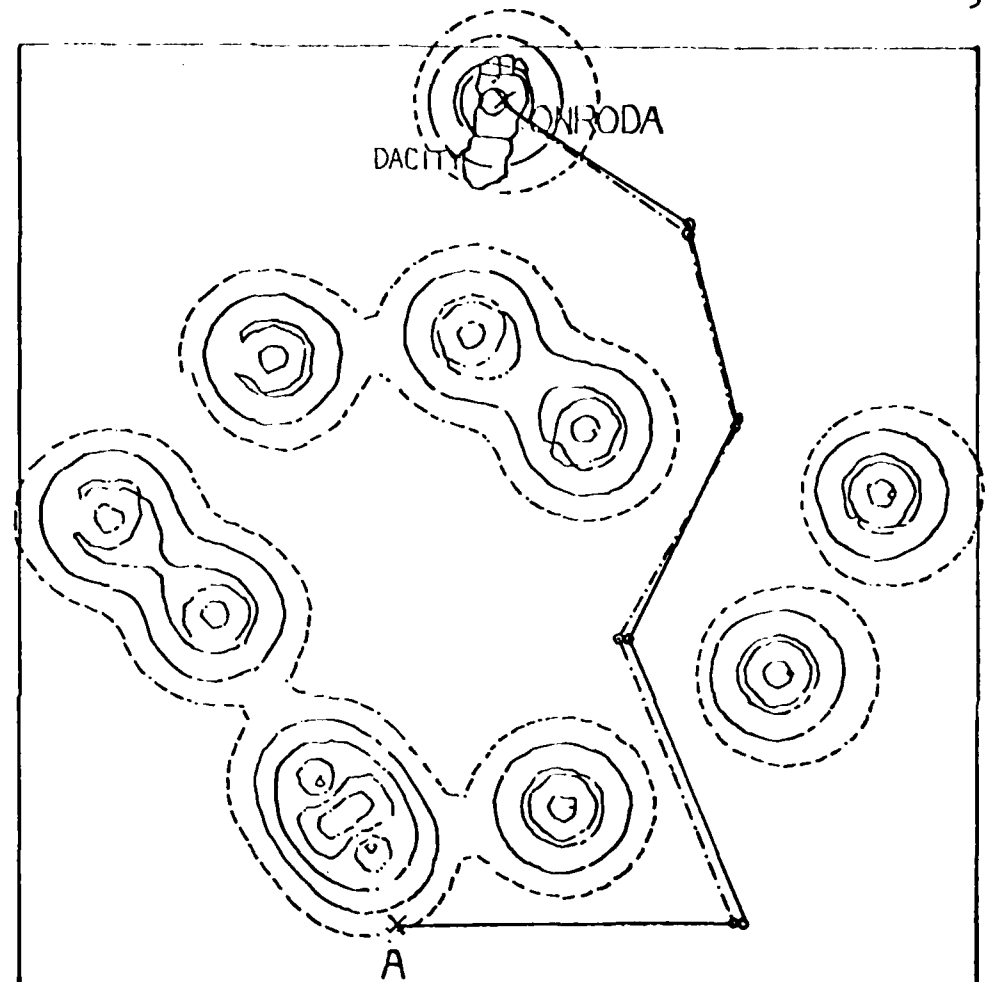
FUNCTION EVALUATIONS 1

PROB = .122
FUEL = 10282.6
UTILITY = 61.36
BEST UTILITY TO DATE = 61.36

Figure 7. Display Appearance After the Operator Has Input His First Starting Point.

MINS
5.3

SPEED (KTS)		
LEG	INIT	CURR
1	380	410
2	270	350
3	270	280
4	520	410
5	600	800



FUNCTION EVALUATIONS 27

PROB = .071
FUEL = 10457.1
UTILITY = 73.73
BEST UTILITY TO DATE = 73.73

Figure 8. Display Appearances After the Operator Has Finished Exploring the Region Around the Starting Point on Figure 7.

in the location of each way point and the speed for each leg. Improvement in the air strike path takes place slowly over many explorations, i.e., trials. An advantage of NP is that it considers all the points in a geographical region instead of just a set of grid points and all speeds instead of just a few. A disadvantage is that the "solution" will be best for the region explored but that better solutions may exist in unexplored regions and the NP technique is unable to direct itself to look in these unexplored regions. In optimization jargon, NP may find a local optimum but not the global optimum.

The subject's purpose is to direct the NP technique to investigate as many reasonable potential solution regions as possible in 15 minutes. As soon as the problem is shown on the display, the subject must decide what region he wants to explore first. He is to pick the region that he thinks is most likely to contain the best solution. He then changes the locations of the way points and speeds prior to starting the NP algorithm. The resultant path and speeds constitute his estimate of the best solution and correspond to the "unaided operator" concept.

When the operator has changed all the way points and speeds to those he wants, he then presses the function button marked "Evaluate/Halt." The NP algorithm will begin to operate, i.e., "Evaluate," using the starting point consisting of the four way points and five speeds. Once the algorithm has begun operating, only the "Evaluate/Halt" button remains lighted, and the only control at the operator's disposal is to halt operation by pushing this button.

The primary indicators that the operator uses to decide whether to halt the algorithm are the displays of the number of function evaluations and the utility of the latest trial solution. In general, a plot of utility versus function evaluations would appear as shown in Figure 9. The subject should stop the algorithm when it reaches the point shown in Figure 9 because there will be little more utility to be gained by letting the algorithm continue. He should then input a new set of way points and speeds and start the algorithm again.

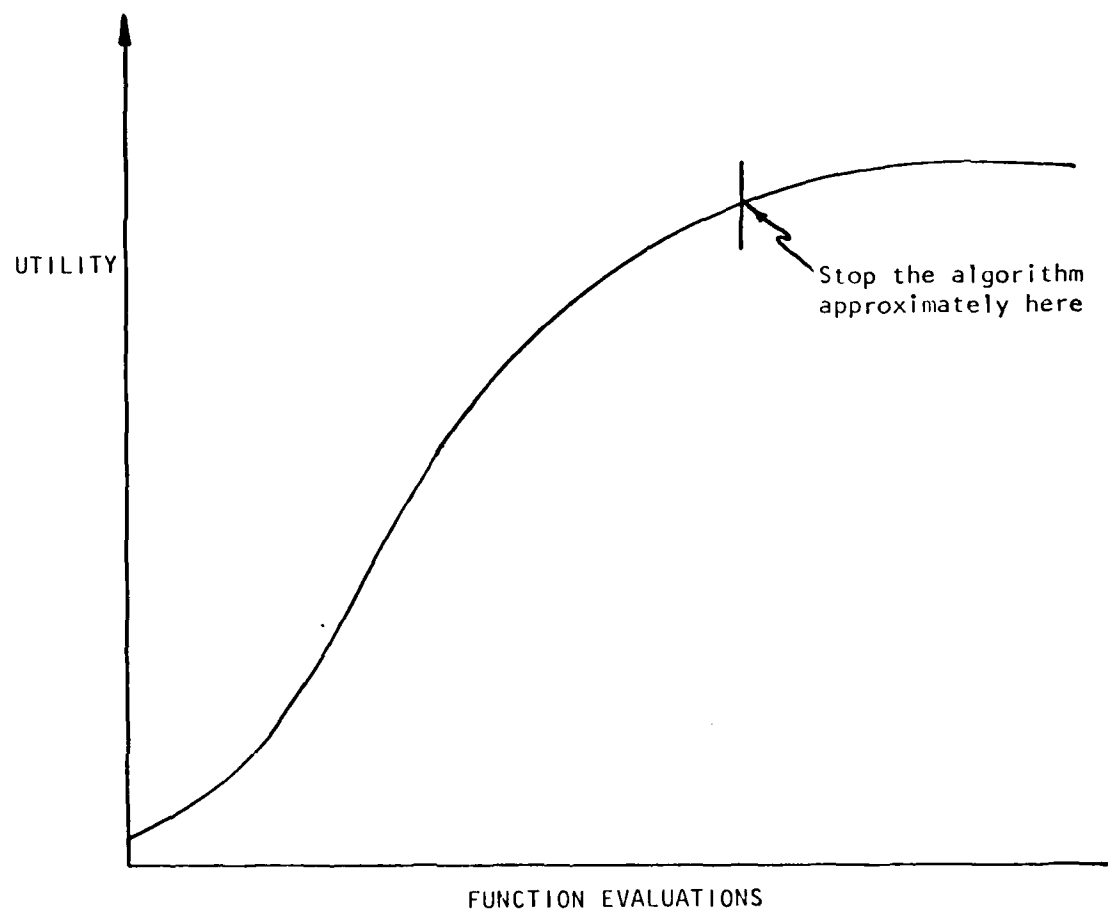


Figure 9. Typical Plot of Utility Versus Function Evaluations.

As the algorithm operates, the operator can see variable-length arrows appearing briefly at each way point. These represent potential changes in the location of a way point being considered by the algorithm. When utility levels off, the magnitudes of changes in the following will also become small:

1. Values of "Prob," i.e., the probability that the air strike will be detected prior to arrival at the target.
2. Value of "Fuel," i.e., the fuel that will be consumed for the latest trial solution.
3. Speed changes indicated on the speed/leg listing at the left of Figure 5.
4. Lengths of arrows appearing at each way point.

While the algorithm is operating on the first set of way points and speeds input by the operator, he should count the number of regions that could reasonably be expected to contain the best path. Dividing 15 minutes by the number of regions to be explored indicates approximately the number of minutes the operator should devote to each region. Depending on the problem, there will be enough time to explore 3, 4, or 5 regions.

At the end of 15 minutes, the computer will have stored:

1. The utility of the path comprised of the first way points and leg speeds entered by the operator.
2. The utility of each best-solution-to-date at the end of each minute, excluding the first minute.

These are the data that are used in the analysis of operator-generated data.

4.0 DESCRIPTION OF THE EXPERIMENTS

4.1 EXPERIMENTAL DESIGN

4.1.1 Hypotheses

Three sets of experimental data were collected and analyzed:

Data Set I: Performance of unaided operators was compared against the performance of the same operators using the IMO aid.

Data Set II: Performance of unaided operators was compared against performance of the same operators using the OAO aid

Data Set III: Performance of operators using the IMO aid was compared against the performance of the same operators using the OAO aid.

The experimental null hypothesis tested in each case was:

Path utilities generated by operators are not significantly different as a function of concept (unaided versus aided or IMO versus OAO), prior experience using the other aid, operators, replications, or their interactions.

4.1.2 Independent Variables

The independent variables for all experiments were:

1. System concepts
2. Prior experience
3. Operators
4. Replications

1. System Concepts. The system concepts which were compared during the experiments were:

1. Unaided¹ versus IMO
2. Unaided versus OAO
3. IMO versus OAO

2. Prior Experience. Eight of the 16 operators worked the IMO problems first and then the OAO problems. The other eight operators did the OAO problems first. Thus, for one set of data, for example, the IMO data, half the data was generated by operators with no prior experience using either the IMO or OAO aid and half was generated by operators with prior experience using the OAO aid.

¹ Recall that the path and speeds chosen by the operator at the beginning of the problem constitute his estimate of the best solution and they correspond to the "unaided operator" concept. (See page 20.)

3. Operators. There were 16 operators. Descriptive information about the operators and their training is given in Subsection 4.2.

4. Replications. Each operator was given a set of 12 problems to be solved using the appropriate procedures (one set of 12 for the IMO aid, another set of 12 for the OAO aid). At the beginning of each problem, the operator recorded his estimate of the best solution. Then he proceeded to use the IMO or OAO procedure. The learning effect was tested by comparing performance on the first six problems against performance on the last six problems. Thus, one replication for operator-generated data consisted of six problems.

4.1.3 Dependent Variables

The dependent variable used in all experiments was normalized utility. The raw data for Unaided Operator were the utilities of the first paths selected by the operator. The raw data for IMO and OAO were the utilities of the best paths found by the operator using IMO or OAO during the fifteen-minute trials. For each problem, these data points were normalized by dividing each value by the highest utility achieved by any operator on that problem. Thus, the experimental hypotheses for data sets I and II were tested by comparing normalized utility of each unaided solution against the normalized utility of the best IMO or OAO solution achieved during each 15-minute trial. The experimental hypothesis for data set III was tested by comparing normalized utility of the best IMO solutions achieved during each 15-minute trial against the normalized utility of the best OAO solutions achieved during each 15-minute trial.

The data collection software also calculated the time average of the best utility to date according to the formula:

$$\bar{U}(T) = \frac{1}{T} \sum_{t=2,3,\dots}^T U(t) \quad (9)$$

where $U(t)$ is the normalized utility at time "t" of the best utility to date.

4.1.4 Problem Variables

The elements that defined a given problem were the adaptation of the ONRODA airstrike scenario map (Subsection 2.1), sensor locations, and strike path start point. The steps described below were taken to make problems nearly equally difficult for the operators.

Each problem used the same number of sensors, namely, ten, and all sensors had the same detection capability. One sensor was always located on ONRODA Island. The remaining nine were positioned by a pseudo-random process. A computer program was written to randomly position the nine sensors subject to two constraints. One constraint was that no pair of sensors could be positioned closer to each other than a certain minimum distance. The other constraint was that all sensors were located below ONRODA Island (see Figure 5). These were realistic constraints since an enemy opposing the air strike would group his forces between ONRODA and the threat and would maintain some minimum spacing between units.

About 50 configurations of sensors generated by the program were examined by the experiment designers. Starting points for the air strike were manually selected so that the largest number of paths having nearly equal utility would result for each of the 50 problems. Then the 24 "best" problems were selected as experimental problems. The basis for selecting the experimental problems was (a) at least three paths having nearly equal utility and (b) no path selection strategy was best for a large majority of the problems. These 24 problems were divided into two sets of 12. Half of the subjects worked Set #1 for IMO and Set #2 for OAO; the other half worked Set #2 for IMO and Set #1 for OAO. Thus, problem difficulty was not treated as an independent variable in the experiment because:

1. Problems were constructed to be nearly equally difficult.
2. Normalization of raw data tends to eliminate whatever differences in problem difficulty remained after the problems were selected.
3. Problem sets worked were balanced between the IMO and OAO concepts.

4.1.5 ANOVA Design

One of the purposes of the experimental work was to determine if the operators would achieve better performance using the IMO and OAO aids for 15 minutes than they would achieve without the aids. A nested factorial, randomized block experiment was conducted. The factors were:

- Concepts (C_i) - 2 levels (Unaided operator and IMO or OAO)
- Prior Experience (P_j) - 2 levels (Half the operators did IMO problems first and OAO next; the other half did OAO, then IMO)
- Operators ($O_{k(j)}$) - 8 levels nested within training; therefore 16 operators total.
- Replications (R_l) - 2 levels (First half of trials and second half).

The other purpose of the experimental work was to compare IMO and OAO. The experiment was the same except that the two levels of concepts were IMO and OAO. There were no designed differences in problem difficulties. Thus, differences in problem difficulties were not treated as a factor. Any spurious differences were mitigated by (a) using normalized data in the analysis and (b) balancing problems across replications and across the IMO/OAO concepts. The model for the normalized dependent variable is:

$$\begin{aligned}
 Y_{ijklm} = & \mu + C_i + P_j + CP_{ij} + O_{k(j)} + CO_{ik(j)} + R_l \\
 & + CR_{il} + PR_{jl} + CPR_{ijl} + OR_{k(j)l} \\
 & + COR_{ik(j)l} + \epsilon_m(ijkl)
 \end{aligned} \tag{10}$$

4.2 OPERATORS AND TRAINING OF OPERATORS

All 16 operators were students from UCLA or Loyola Marymount majoring in engineering, science, or mathematics. Operator training for use of each aid, i.e., the IMO aid and the OAO aid, was conducted in three phases: orientation, demonstration, and exercise using training problems. Orientation for each aid began with reading the training materials developed for that aid. The training materials treated the following topics:

- Purpose of the experiment
- Representation of sensor detection capability on the display
- The utility function
- Characteristics of the optimization technique used (IMO or OAO)
- Operation of the aid
- Example of a problem worked out (nineteen figures, with text comments and guidelines accompanying each figure).

The training materials are in Appendices B and C.

After each operator read the training materials, he conferred with one of the ISC staff members who designed the experiment. Operator questions were answered during this conference and the ISC staffer verbally tested the operator's understanding of the problem situation and use of the aid. The ISC staffer then demonstrated the use of the aid and focused on the discussion of strategy in the training materials. Then the operator worked eight problems at the display. Questions that arose during these problems were answered by an ISC staff member with experience using the aid. The operator began his experimental trials after this training period. No further training was given during the trials. The average training time across operators was about four hours for each aid.

4.3 EXPERIMENTAL PROCEDURE

The experimental team consisted of the test director and an operator. Each operator was assigned a unique identification code, and the sequence of the twelve problems corresponding to that code was stored in the computer. The test director scheduled the software and entered the operator's code. That procedure "brought up" the operator's next (uncompleted) problem on the display. At the beginning of the problem, the operator entered his best estimate of the solution as described in Subsections 3.1 (for the IMO aid) or in 3.2 (for the OAO aid). The operators were told to watch the displayed clock time as they had two minutes in which to enter their first estimate. Based on the observation of the test directors, no operator had problems with this time limit after one or two training trials.

Feedback to the operator on his performance was provided throughout each trial. At the end of each minute, the computer calculated and displayed the operator's time averaged performance. The display of this value for all minutes from minute 2 onward was located just above the picture of the scenario.

The test director remained on call during each trial to monitor the trials, troubleshoot any equipment malfunctions or operator-induced problems in entering path data, and to bring up the next trial once the previous trial was completed. The test director spent part of the time in the computer and display facility where the operator worked the problem and the remaining time in an adjacent room. Operators normally did two or three trials in a row before taking a break. Multiple trials were permitted because operators did not experience fatigue after as many as three sequential trials.

5.0 RESULTS

5.1 ANALYSIS OF UNAIDED OPERATOR AGAINST IMO DATA

A four-way analysis of variance (ANOVA) was performed on the normalized path utility data generated by operators in the "unaided" mode and with the IMO aid. The IMO data point used from each trial was the utility of the best path found during the trial. (Strictly speaking, the unaided mode is not completely unaided since the operator has the use of the contours of composite detection capability of enemy sensors.) Preliminary tests on the model were made at the 20% significance level, and pooling procedures were applied (Reference 5). After pooling, the ANOVA results were as shown in Table 2.

As expected, the operators using the IMO aid did significantly better than without it (unaided mode). The average improvement across all operators and trials was 11%, i.e., from a normalized utility of 88.75 to one of 98.13 on a scale of 100 possible points. There is a significant effect of prior experience using the OAO aid; this must be examined together with the significant interaction between this prior experience and the concepts effect (unaided vs. IMO). This interaction is plotted in Figure 10. The figure, together with further statistical tests, shows that operators having prior experience with the OAO aid did significantly better by 7.5% in the unaided mode than operators without such experience. However, prior experience with OAO had no significant effect on performance using the IMO aid. Also, operators who had previously used the OAO aid showed a lesser (7%), but still significant, improvement between unaided and IMO scores than did the operators who had no prior experience (14% improvement).

Performance was significantly different between operators, and there was also a significant interaction between this effect and concepts (unaided vs. IMO). Operators' normalized scores in the unaided mode ranged from 65.9 to 96.1; normalized scores using IMO varied only from 95.1 to 99.7.

Table 2. Unaided Operator vs. IMO ANOVA (with pooling)

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Square	F _{obs}
Concepts (C)	1	8,383.	8,383.	$F_{351}^1 = 77.44^*$
Prior Experience (P)	1	1,115.	1,115.	$F_{351}^1 = 10.30^*$
Operators (O)	14	5,503.	393.1	$F_{351}^{14} = 3.63^*$
C X P	1	870.5	870.5	$F_{351}^1 = 8.04^*$
C X O	14	3,599.5	257.1	$F_{351}^{14} = 2.38^*$
Replications (R)	1	10.	10.	$F_{351}^1 = 0.09$
Pooled Error	351	37,997.	108.25	
TOTALS	383	57,478.		

* $\alpha < .05$

Table 3. Unaided Operator vs. OAO ANOVA (with pooling)

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Square	F _{obs}
Concepts (C)	1	6,492.5	6,492.5	$F_1^1 = 20.74$
Prior Experience (P)	1	524.5	524.5	$F_{350}^1 = 9.46^*$
Operators (O)	14	1,586.5	113.3	$F_{350}^{14} = 2.04^*$
C X P	1	286.	286.	$F_{350}^1 = 5.16^*$
C X O	14	905.5	64.7	$F_{350}^{14} = 1.17$
Replications (R)	1	435.	435.	$F_{350}^1 = 7.84^*$
C X R	1	313.	313.	$F_{350}^1 = 5.64^*$
Pooled Error	350	19,409.	55.45	
TOTALS	383	29,952.		

* $\alpha < .05$

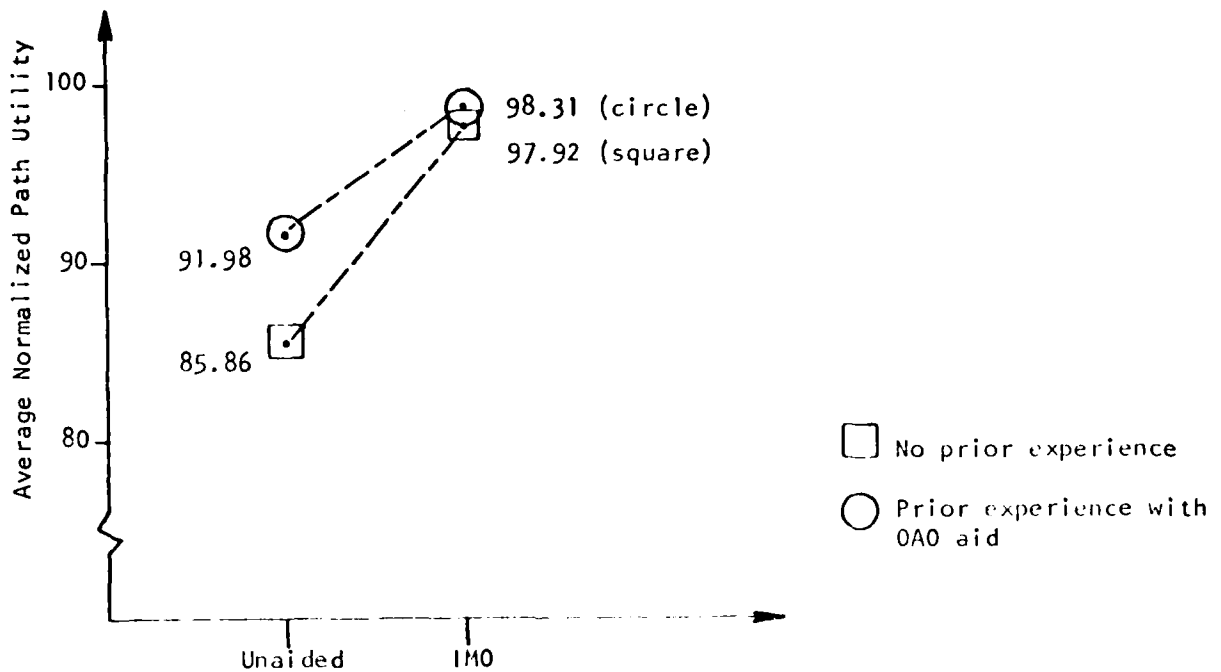


Figure 10. Interaction Between Prior Experience and Concepts in the IMO Experimental Results.

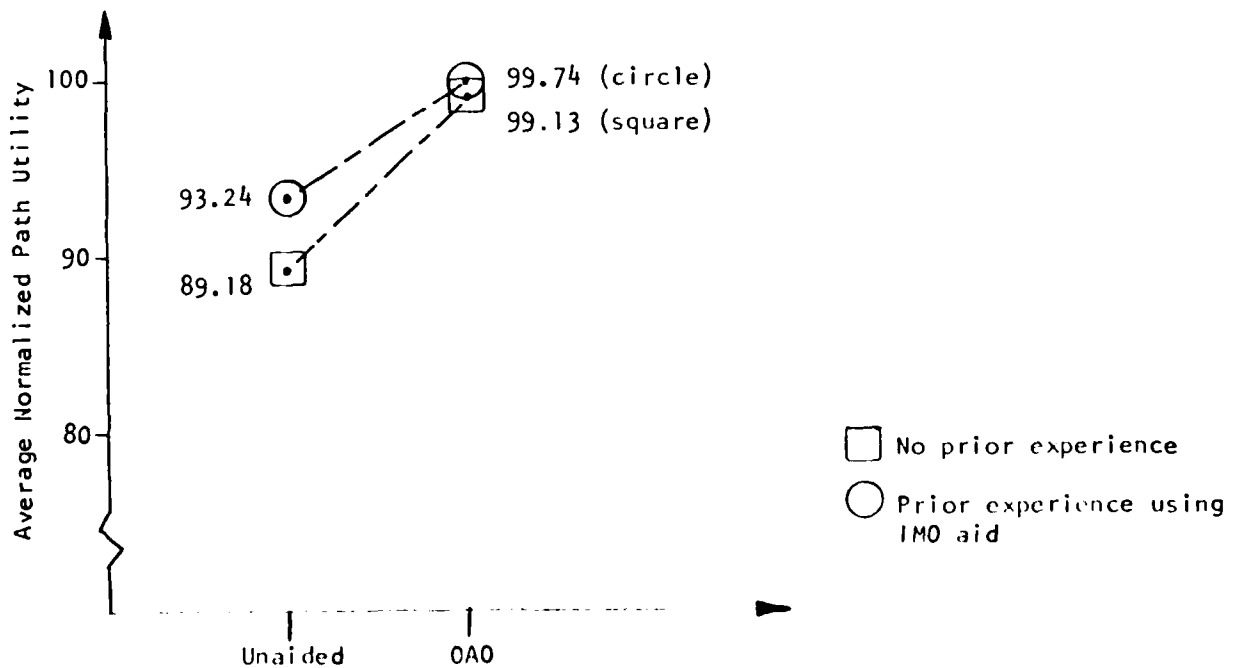


Figure 11. Interaction Between Prior Experience and Concepts in the OAO Experimental Results.

Thus, the aid served as an "equalizer." It enabled operators having relatively low scores in the unaided mode to do as well when using the IMO aid as those who had relatively high scores without the aid. Inspection of the results also showed that the operators with the highest unaided scores in fact did not improve significantly using the IMO aid. However, as seen above, these generally were the operators with prior experience using the OAO aid.

5.2 ANALYSIS OF UNAIDED OPERATOR AGAINST OAO DATA

A four-way ANOVA was performed on the normalized path utility data generated by operators in the unaided mode and with the OAO aid. The OAO data point used from each trial was the utility of the best path found during the trial. After pooling, the ANOVA results were as shown in Table 3.

The operators using the OAO aid, as expected, did significantly better than without it (unaided mode). The average improvement across all operators and trials was 9.0%, i.e., from a normalized utility of 91.20 to one of 99.43 on a scale of 100 possible points. There is a significant effect of prior experience; this must be examined together with the significant interaction between this prior experience and the concepts effect (unaided vs. OAO). This interaction is plotted in Figure 11. The figure, together with further statistical tests, shows that operators with prior experience with the IMO aid did significantly better (by 4.6%) in the unaided mode than operators without such experience. However, prior experience with IMO had no significant effect on performance using the OAO aid. Also, operators who had previously used the IMO aid showed a lesser (7%), but still significant, improvement between unaided and OAO scores than did the operators who had no prior experience (11% improvement).

There was a significant replications effect; this must be examined together with the significant interaction between this effect and concepts (unaided vs. OAO). This interaction is plotted in Figure 12. Operators in

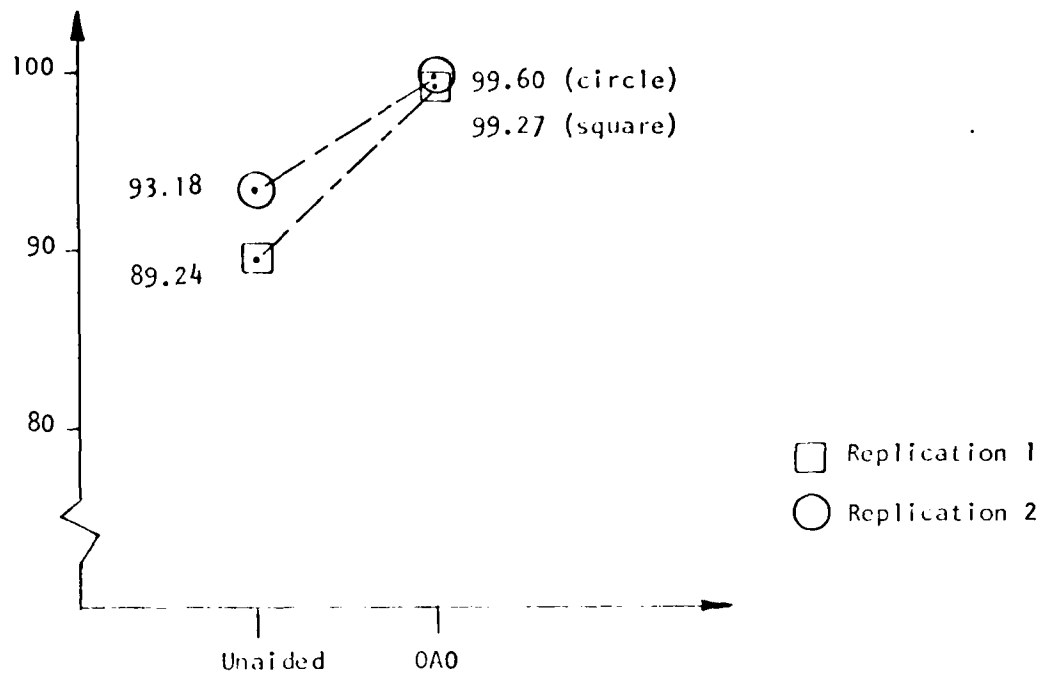


Figure 12. Interaction Between Replicating and Concepts in the OAO Experimental Results

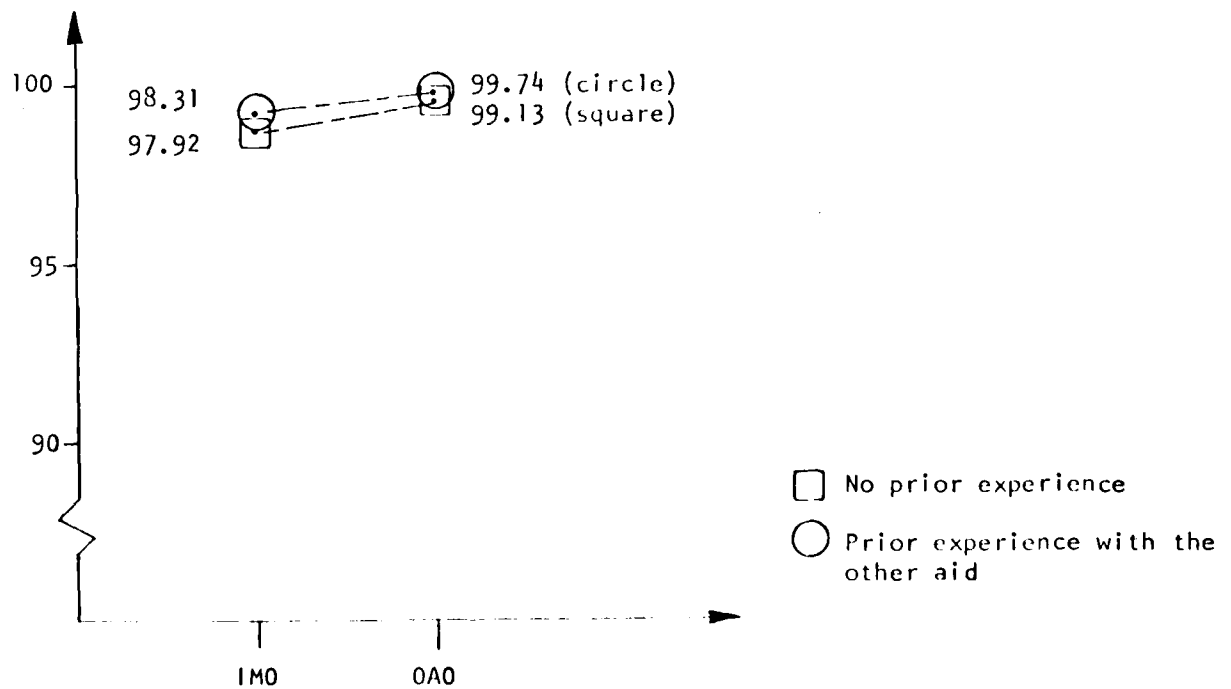


Figure 13. Interaction Between Prior Experience and Concepts in the IMO vs. OAO Experimental Results.

the unaided mode did significantly better (by 4.4%) during replication 2 than during replication 1. However, there was no significant replication effect in OAO mode. Also, replication 2 showed a lesser (7%), but still significant, improvement between unaided and OAO scores than during replication 1 (11% improvement).

Performance was significantly different between operators. Once again, this difference was greater in the unaided mode (range 82.9 to 97.2) rather than in OAO (range 95.7 to 100.0). Thus, the OAO aid like the IMO aid, served as an "equalizer."

5.3 ANALYSIS OF IMO AGAINST OAO

A four-way ANOVA was performed on the normalized path utility data generated by operators using the IMO aid and the OAO aid. After pooling, the ANOVA results were as shown in Table 4.

Operators using the OAO aid did slightly, but significantly, better than those using the IMO aid. The average improvement across all operators and trials was 1.3%, i.e., from 98.13 to 99.43 on a scale of 100 possible points. There was a significant interaction between this effect and prior experience. In order to examine this interaction, prior experience was redefined, from "IMO first" vs. "OAO first," to "no prior experience" vs. "prior experience with the other aid." With this redefinition, the interaction disappears. Figure 13 is a plot of the results of this analysis. It shows a very similar improvement (about 1.3 points) for OAO over IMO, both with and without prior experience with another aid; there was also a very slight (about 0.5 points) but significant improvement for both IMO and OAO if the operator had prior experience with the other aid.

Performance was, again, significantly different between operators, with significant interactions this time with both concepts and replications. The operators/concepts interaction means that some operators did slightly better with one aid than the other, relative to the average performance

Table 4: IMO vs. OAO ANOVA (with pooling)

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Square	F _{obs}
Concepts (C)	1	166.69	166.69	$F_{337}^1 = 31.30^*$
Prior Experience (P)	1	1.096	1.096	$F_{337}^1 = 0.206$
Operators (O)	14	273.11	19.508	$F_{337}^{14} = 3.663^*$
C X P	1	23.863	23.863	$F_{337}^1 = 4.481^*$
C X O	14	136.21	9.729	$F_{337}^{14} = 1.827^*$
Replications (R)	1	13.812	13.812	$F_{337}^1 = 2.594$
O X R	14	153.17	10.941	$F_{337}^{14} = 2.054^*$
Pooled Error	337	1,794.7	5.326	---
TOTAL	383	2,562.7		

* $\alpha < .05$

over all operators for each aid, i.e., some operators "took" to one aid slightly more than the other. The operators/replications interaction means that some operators improved slightly on the second replication, while others did slightly worse, although the overall replications effect was not significant.

The performance of operators using IMO and OAO was also compared as a function of time by considering the experiment from minute 2 to minute 15 (instead of just at minute 15 as in the analysis of variance above). Figure 14 shows a plot of best-to-data normalized utility versus time for IMO and OAO. The results of the previous OAO experiment (Reference 2) are shown for comparison, as are the results for the unaided operator (averaged across all IMO and OAO trials for this experiment). The improvement given by OAO over IMO decreases from 3.9% at minute 2 to 1.3% at minute 15. Figure 15 shows a plot of time-averaged scoring rule (Equation 9) versus time for IMO and OAO. The improvement given by OAO over IMO again decreases from 3.9% at minute 2 to 2.1% at minute 15.

5.4 FOLLOW-UP QUESTIONNAIRE

Each operator was given a questionnaire upon completion of his experimental trials. The questions were:

1. If you could change the IMO aid (no optimizer) or its operations, how would you do it?
2. If you could change the OAO aid (with optimizer) or its operations, how would you do it?
3. If you could have either of the changed aids to solve the problems, which do you think would enable you to do the better job?

The most frequent response to the first two questions was the recommendation that the aids include a capacity for increasing the number

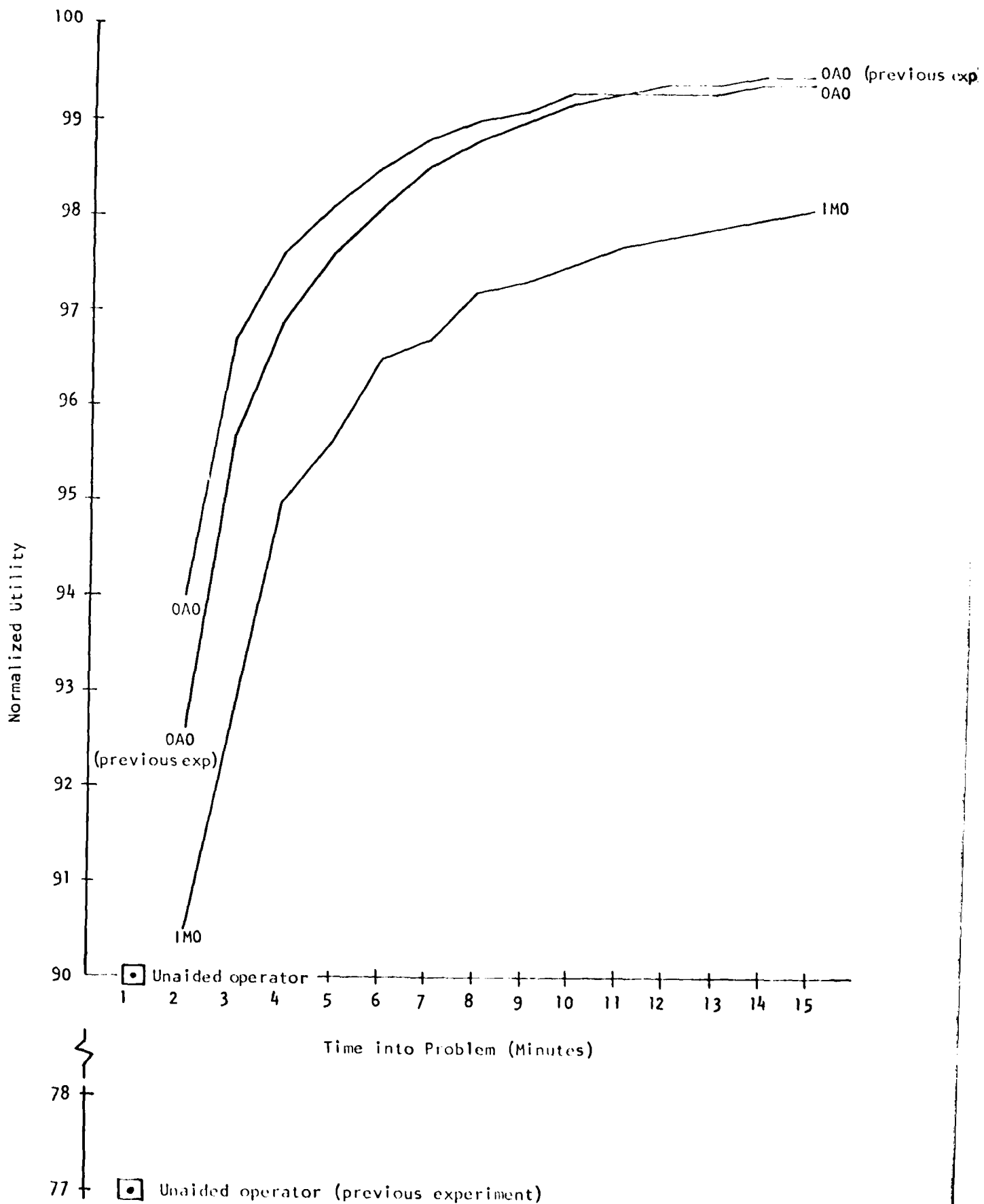


Figure 14. Best-to-Date Normalized Utility Versus Time for OAO and IMO.

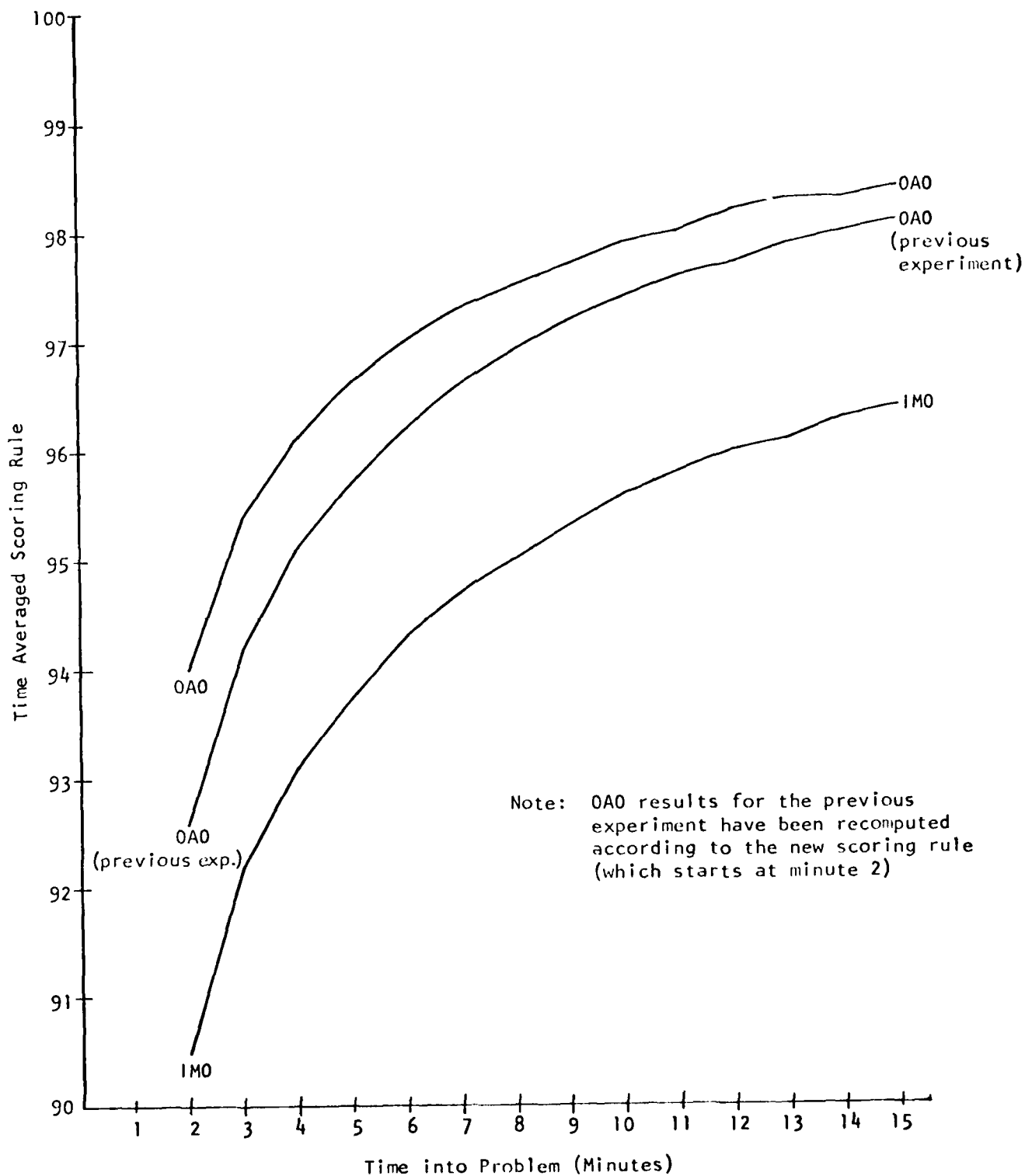


Figure 15. Time Averaged Scoring Rule Utility Versus Time for OAO and IMO.

of way points. (This was not done in the experiment because it would have added an uncontrolled variable.) Another frequent recommendation was that the speeds for the best-path-to-date be displayed alongside that path on the screen. Three subjects recommended the second-best-solution-to-date be held on the screen in addition to the best solution. One subject wanted more detection capability contours to be displayed in order to increase resolution. Another subject noted that most of the optimum speeds for legs were less than the default speed of 600 knots and recommended that the default speed be changed to a lower number.

All the subjects preferred the OAO aid. The reasons commonly given were:

1. The OAO aid enabled the operator to explore solution regions more fully than the IMO aid.
2. The OAO aid was less tiring to use than the IMO aid.

6.0 DISCUSSION OF RESULTS

6.1 SUMMARY OF EXPERIMENTAL FINDINGS

The analyses of "unaided" operator mode versus performance with the IMO and OAO aids showed similar results. In each case, there was about a 10% average improvement across all operators and trials between the unaided and aided modes. For IMO, the improvement was from 88.75 (unaided) to 98.13 (aided); for OAO, from 91.20 to 99.43. In each case, operators who had prior experience with aid #1 (IMO or OAO) performed better (by about 6%) in the unaided mode of aid #2 (OAO or IMO, respectively) than those with no prior experience. There was a much wider range of performance in the unaided mode than in either aided mode; thus, both aids served as "equalizers" of operator performance differences.

The analysis of IMO versus OAO showed a very slight advantage to OAO: an average of 1.3 points, or 1.3% (from 98.13 to 99.43 on a scale of 100). This difference was independent of prior experience with the other aid, although prior experience with aid #1 (IMO or OAO) did raise the average score on aid #2 (OAO or IMO, respectively) by about 0.5 points. Examining performance over the entire 15 minutes of an experimental trial, the slight advantage of OAO over IMO decreases from 3.9% at minute 2 (94.0 versus 90.5) to 1.3% at minute 15 (as noted above).

6.2 PERFORMANCE DIFFERENCES IN THE UNAIDED MODE

Average performance in the unaided mode (90.0 points) was considerably better in the current study than in the earlier study of OAO, 77.2 points (Ref. 2). At least two factors may have contributed to this improvement. One factor is the difference in the elapsed time between starting a trial and recording the first best-to-date path utility used in calculating the time-averaged utility for the scoring rule. The other factor is a qualitative difference in training given to operators. These factors are discussed below.

In the previous experiment, scoring rule results were calculated at the end of each trial minute beginning with the end of the first minute. It was difficult to make all the desired inputs for the first solution, i.e., the "unaided-operator" solution, within a single minute. An operator who was unable to make all the desired inputs for the unaided operator solution within the first minute had two choices:

1. Continue making inputs to the unaided-operator solutions after the end of the first minute. In this case the unaided-operator solution would be better than it would have been if the operator had pressed the "Evaluate" button before the end of the first minute. However, the value for the scoring rule would have suffered because the computer would have used a zero for the path utility at the end of the first minute.

2. Stop making inputs to the unaided operator solution before the end of the first minute and press the "Evaluate" button. In this case the unaided operator solution suffered but the value for the scoring rule gained because the computer had a solution at the end of the first minute and thus the scoring rule had a non-zero value at the end of the first minute.

A substantial number of operators in the previous experiment usually made the second choice.

In the current experiment, scoring rule results were calculated at the end of each trial minute beginning with the end of the second minute. Since everyone was able to make all desired inputs to the unaided operator solution within two minutes, operators were not faced with the two choices described above. The additional time available to input the unaided solution presumably enabled the operators to select better unaided solutions than was possible in the previous experiment.

The training materials for the current study include a more concrete discussion of path selection strategy, including typical near-optimal speeds for sample path segments. This was re-emphasized during demonstration of the aids and early training. The IMO aid required careful attention to path selection strategy; even those operators who used the OAO aid first were aware that they needed to follow the optimizer closely in order to learn proper strategy to use later with IMO. All this attention to strategy may also have improved performance on the unaided-operator estimates of optimal paths and speeds.

Given the improved score of 90.0 points in the unaided mode, it is not surprising that the performance difference between the IMO and OAO aids was small at minute 2 and still smaller at minute 15. In addition, it was a relatively simple visual recognition task to place the way points on a route through the valleys of the detection capability contour map in order to minimize the cumulative detection capability. There were no visual aids to help determine the best speeds on each leg. However, the guidance on this given in training, together with the fact that the utility function was less sensitive to errors in speed decisions than it was to errors in way point placement, made the lack of visual aids here less important.

6.3 SOFTWARE COSTS ASSOCIATED WITH IMPLEMENTING AND USING THE OPTIMIZING ALGORITHM

In general, the following software costs are associated with implementing and using an optimizing algorithm to solve a complex problem:

1. Devising and implementing a unique algorithm or selecting an existing algorithm.
2. Connecting the optimizing algorithm to other routines that are part of a larger software system.
3. Computer memory storage space occupied by the optimizer.
4. Computer central processing unit (CPU) time required to reach a solution.

The algorithm used in the OAO NP aid was selected from a publication to which ISC subscribes, the "Collected Algorithms of ACM."¹ As implemented in the program, it consists of 156 lines of FORTRAN; 38 of the 156 lines are program documentation "comments."² There are two aspects to connecting the algorithm to other routines in the program. One is that there are seven "calls" in the optimization routine to other routines in the overall software system. Part of the effort in implementing the optimizing algorithm is inserting the "calls" in the appropriate places. The other implementation task for the OAO NP algorithm was writing about 60 lines of code that enable the display to show display details that are not present (and not applicable) for the IMO aid. Thus, the software costs associated with implementing the algorithm were a very small part of generating the code for the entire OAO aid.

The OAO program is approximately 29,880 decimal words long and the IMO program is approximately 28,480 decimal words. Since these programs are identical except for the presence of the optimizer in the OAO aid, the difference of 1,400 words (5.1% of IMO length) is attributable to the optimizer. A difference between IMO and OAO is the amount of CPU use for each. OAO uses the CPU continuously between the time the operator activates "Evaluate" and, later, "Halt." With IMO, the CPU is only used to calculate the values of cumulative probability of detection, fuel remaining, and the utility function once for each set of waypoint and/or speed changes input by the operator. The importance of this difference depends on the amount of OAO use on a computer system that would have many other jobs to run.

If a choice is to be made between the IMO and OAO aids developed for the current experiment, it should be based on comparisons of performance, ease of use, and software costs. The performance comparison favors OAO, but only by a slight amount. OAO is much preferred by operators because it is less tedious to use. The authors believe that the tradeoff between

¹ Association for Computing Machinery.

² The entire OAO program is 1,607 lines of FORTRAN. This includes the data collection software but not the data analysis software.

the extra software costs associated with OAO and the performance and ease of use benefits of OAO would favor selection of the OAO aid.

In general, comparisons between OAO- and IMO-type aids for other problems of comparable complexity will produce the same results found here, namely, the OAO-type aids will:

1. Provide better performance than an IMO-type aid. (However, for problems having much greater complexity, the performance margin favoring OAO might be much larger or the performance comparison might even favor an IMO-type aid.)
2. Be easier to use than an IMO-type aid.
3. Have a higher software cost than an IMO-type aid.

The tradeoff among these differences will usually favor the OAO-type aid whenever the optimizing algorithm is a standardized, already-coded algorithm that only needs to be connected to the main program with a few "call" statements. However, if a tailor-made optimizer must be developed for a particular problem, then the results of a tradeoff might change the choice between an IMO-aid and an OAO-type aid.

6.4 RECOMMENDATIONS

The real world of air strike planning is more complex than the problem used to compare the OAO and IMO concepts. Important real-world considerations are listed below:

Problem Factors

- capabilities and location of enemy sensors
- capabilities and locations of enemy missiles
- capabilities of enemy fighters and locations of their bases
- capabilities and locations of enemy anti-aircraft guns
- change in location of enemy defense forces during the flight time of the air strike

Decision Dimensions

- x,y locations of path legs
- speed on each leg
- altitude on each leg
- when to use on-board jamming equipment

Utility Dimensions

- Expected damage to strike aircraft due to enemy defenses
- Distance between target and strike aircraft when cumulative detection probability exceeds x%
- Probability of strike mission success
- Fuel remaining

The IMO and OAO aids described in this report account for only a few of the dimensions. Consequently, we recommend the following question for further study:

Is the small magnitude of the performance difference between IMO and OAO observed in the current study due to the relative simplicity of the problems to be solved?

This question may be put another way:

How will the difference in operator performance using IMO- and OAO-type aids change as the dimensions of the problem, decision, and utility function increase?

Performance data from aids and experiments designed to answer this question could be important evidence supporting a decision by R&D program managers to concentrate future efforts on developing one of the aiding concepts in preference to the other.

REFERENCES

1. Irving, G. W., et. al. Experimental Investigation of Sketch Model Accuracy and Usefulness in a Simulated Tactical Decision Aiding Task. Report No. 215-3, Integrated Sciences Corporation, Santa Monica, California, May 1977.
2. Walsh, D. H. and Schechterman, M. D. Experimental Investigation of the Usefulness of Operator Aided Optimization in a Simulated Tactical Decision Aiding Task. Report No. 215-4, Integrated Sciences Corporation, Santa Monica, California, January 1978.
3. Payne, J. R. and Rowney, J. V. ONRODA Warfare Scenario. SRI/NWRC-RM-83, Stanford Research Institute, Menlo Park, California, June 1975. AD# AO 16625.
4. Zehna, Park W. (Ed.) Selected Methods and Models in Military Operations Research. Department of Operations Research and Administrative Sciences, Navy Postgraduate School, Monterey, California, 1971.
5. Winer, B. J. Statistical Principles in Experimental Design, 2nd Edition. New York, McGraw Hill, 1971.
6. Powell, M.J.D., "An Efficient Method of Finding the Minimum of a Function of Several Variables Without Calculating Derivatives." The Computer Journal, Vol. 7, 1964, pp. 155-162.

APPENDIX A:

NONLINEAR PROGRAMMING OPTIMAL PATH SOLUTION

APPENDIX A: NONLINEAR PROGRAMMING OPTIMAL PATH SOLUTION

The method chosen for the nonlinear programming optimization was developed and proposed by H.H. Rosenbrock (Reference 3). It is an attractive method to use for this application because it does not require the calculation of derivatives. It is fairly efficient in the number of function evaluations needed while at the same time being able to handle a wide variety of function types.

The derivative-free characteristic is necessary in this application due to the nature of the function being optimized, the utility function in this case. The function is not explicitly expressed in the variables that are being controlled. The utility is a direct function of performance measures (e.g., penetration ratio) and cost. The control variables on the other hand include sensor types, number of sensors and location. Once these are specified the simulator (NIBS) calculates the performance figures to be used in the evaluation of this utility function. The derivatives of such a function clearly cannot be analytically obtained. This fact eliminated from consideration all the optimization methods which require derivative calculations (conjugate gradient, Newtons, Fletcher-Powell, etc.).

A number of derivative-free methods exist. These include Rosenbrock's, the simplex method of Himsworth, Spendley and Hext, Smith's method based on a conjugate direction, and of course, simple univariate search (Reference 6). Any of these might be suitable for the job. Rosenbrock's method was chosen because it had the added flexibility of allowing the introduction of constraints on the controlled variables. These functions could be defined separately from the utility function that would control the region through which search was permitted. Although constrained optimization is not a requirement, it was felt that it might be needed depending on the global behavior of the utility function. This added flexibility was deemed sufficient cause to select Rosenbrock's method as the candidate optimization scheme.

Rosenbrock's method is an extension of the univariate search method. In univariate search the minimum of a function $u(x_1, x_2, \dots, x_n)$ is found by searching along each of the x_i directions in turn. After reducing u as far as possible with each variable, the procedure moves on to the next variable in a cyclical fashion. This method can bog down on elongated functions with deep troughs. This is because the search directions are fixed and do not change as a result of progress through the function. The method developed by Rosenbrock is meant to eliminate this fault without adding a great deal of complexity or the need for derivative calculations. The method basically consists of finding two factors: (1) Length of Step and (2) Direction of Step. Using these two factors according to the algorithm proposed by Rosenbrock, function minimization can be accomplished in an efficient manner on a wide variety of function types.

The simplest problem is to decide the length of step to be taken in the desired direction, assuming this direction to be known. The principle adopted is to try a step of arbitrary length e . If this succeeds, e is multiplied by $\alpha > 1$. If it fails, e is multiplied by $-\beta$ where $0 < \beta < 1$. "Success" here is defined to mean that the new value of u is less than or equal to the old value for a minimization problem. Thus if e is initially so small that it makes no change in u , it is increased on the next attempt. Each such attempt is called a "trial."

The remaining factor is to decide when and how to change the directions $\underline{\xi}$ in which the steps are taken. The method uses n orthogonal directions $\xi_1, \xi_2, \dots, \xi_n$. One trial of the univariate type is made in each of the n directions in turn. This is done until at least one trial is successful in each direction, and one has failed in each direction. It will be noticed that a trial must in the end succeed because e becomes so small after repeated failures that it causes no change in u . The set of trials made with one set of directions, and the subsequent change of these directions, is called a "stage."

The method chosen for finding the new directions of ξ was the following. Suppose that d_1 is the algebraic sum of the lengths of all the successful steps e_1 , in the direction ξ_1 , etc. Then let

$$\left. \begin{aligned} A_1 &= d_1 \xi_1^0 + d_2 \xi_2^0 + \dots + d_n \xi_n^0 \\ A_2 &= d_2 \xi_2^0 + \dots + d_n \xi_n^0 \\ &\dots \dots \dots \\ A_n &= d_n \xi_n^0 \end{aligned} \right\} \quad (A.1)$$

Thus A_1 is the vector joining the initial and final points obtained by use of the vectors $\xi_1^0, \xi_2^0, \dots, \xi_n^0$, A_2 is the sum of all the advances made in directions other than the first, etc.

Orthogonal unit vectors $\xi_1^1, \xi_2^1, \dots, \xi_n^1$, are now obtained in the following way:

$$\left. \begin{aligned} B_1 &= A_1 \\ \xi_1^1 &= B_1 / |B_1| \\ B_2 &= A_2 - A_2 \cdot \xi_1^1 \xi_1^1 \\ \xi_2^1 &= B_2 / |B_2| \\ &\dots \dots \dots \\ B_n &= A_n - \sum_{j=1}^{n-1} A_n \cdot \xi_j^1 \xi_j^1 \\ \xi_n^1 &= B_n / |B_n| \end{aligned} \right\} \quad (A.2)$$

No ambiguity is likely to arise, since the method used ensures that no d can be zero. It is of course possible that one or more of the d are so small that they are lost in the summations of equations (A.1), but this is unlikely in practice. The result of applying equations (A.1) and (A.2) several times is to ensure that ξ_1 lies along the direction of fastest advance, ξ_2 along the best direction which can be found normal to ξ_1 , and so on.

The numerical work of developing this process was carried out to determine appropriate values for α and β . In addition, tests were run on a variety of functions in comparison with other available methods. As a result of testing Rosenbrock selected the values $\alpha = 3$, $\beta = 0.5$ for use in his method. Using these values he found that his method was not significantly

slower than the available alternatives in simple problems. In difficult problems he claims it may be a good deal faster. It is well adapted to automatic calculation, and is not easily upset by minor irregularities such as occur in asymmetrical ridges. The method permits the introduction of constraints into the minimization problem.

APPENDIX B:
TRAINING MATERIALS FOR IMO

I. INTRODUCTION

Integrated Sciences Corporation is conducting a study for the Office of Naval Research that investigates ways to allocate functions between humans and computers so that their respective strengths are best used. The portion of the study in which you are participating seeks to determine to what extent, if any, selected optimization techniques can aid and thus improve the performance of a human operator when applied to a naval tactical decision problem. For now, you will be working with a computer display, but without the aid of the optimizer. In other words, you will do any optimization yourself. We call this "iterative manual optimization," or IMO for short.

Your role in the experiment is to act as the member of a Naval Task Force Commander's staff who is planning a tactical airstrike against the airfield on a place called ONRODA Island. Your Naval Task Force consists of aircraft carriers, their squadrons of aircraft, and escort ships. They are located approximately at the point marked with an X in Figure B1. About ten enemy ships are located in a region between your Task Force and ONRODA. Important parts of air strike planning are (a) deciding the path that your aircraft will take to get to the target and (b) strike aircraft speeds along the legs of the path. As air strike planner, you must be concerned about these two factors.

1. The probability that your aircraft will be detected before they reach the target. If they are detected before reaching the target, the enemy will be at maximum readiness to repel the air strike there. The enemy ships between your Task Force and ONRODA Island have radar that could detect your aircraft. However, the enemy ships themselves have no interceptor aircraft nor do they have guns or missiles that would be effective against your aircraft.

2. Amount of fuel left aboard your aircraft when they reach the target. It is desirable to maximize the fuel left in order to engage or avoid enemy interceptor aircraft over the target or to attack secondary targets once the primary target, ONRODA airfield, has been destroyed. Your job is to help the computer come up with the best airstrike plan between the task force and the target, within a specified time limit.

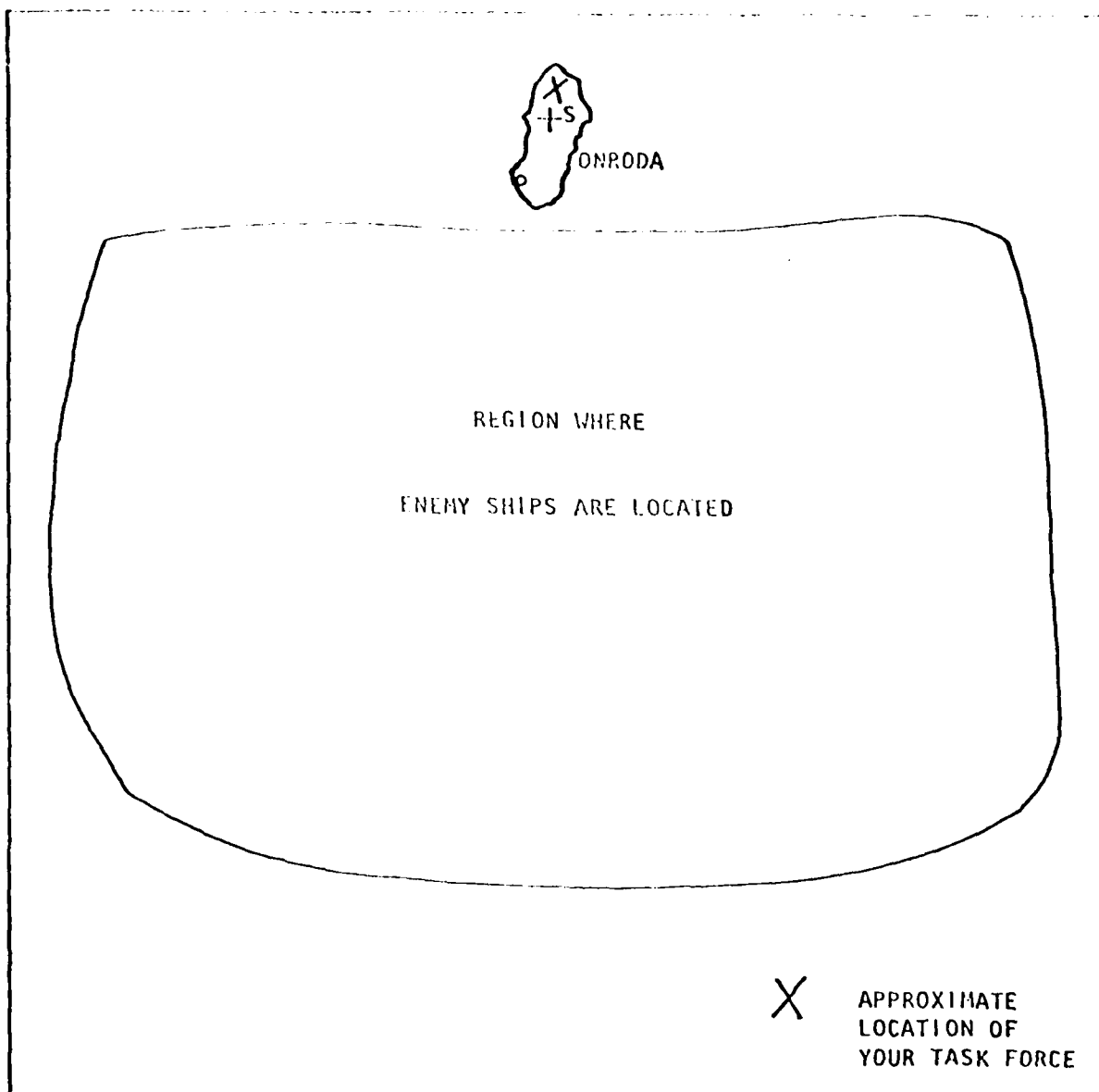


Figure B1. Map of Area of Interest.

The best air strike plan minimizes the probability of the aircraft being detected by the radars and, at the same time, maximizes the fuel remaining upon arrival at the target so that the enemy fighter aircraft can be engaged or evaded.

The purpose of this material is to acquaint you with the:

1. Detection ability of multiple enemy radars when there is overlapping detection coverage between radars in proximity to each other
2. Means of measuring the goodness of an air strike plan
3. Characteristics of the nonlinear programming optimization technique.

The training goals are to:

1. Develop expertise in using the equipment
2. Develop a feel for the best way to work with the computer to find the best air strike paths and speeds.

In training you will do eight problems; experimental data collection will then be done for twelve problems. Thus, you will do a total of 20 problems. Each problem will last 15 minutes.

A. REPRESENTATION OF ENEMY RADAR DETECTION CAPABILITY

The capability of a single enemy radar to detect your aircraft is represented by concentric circles around the radar location. Detection capability is the same at all points on each circle and is a specified percentage of the peak detection capability of the radar. (See Figure B2.) Notice that as you go along a radial line toward the center of the concentric circles, detection capability increases up to 90% of the peak level. The peak occurs between the two 90% circles, and detection capability decreases from the peak as you get closer to the radar location. Thus detection capability may be visualized in three dimensions as a volcano with a rim and a crater in the center of the volcano. The "Detection volcano" is centered on the radar's position.

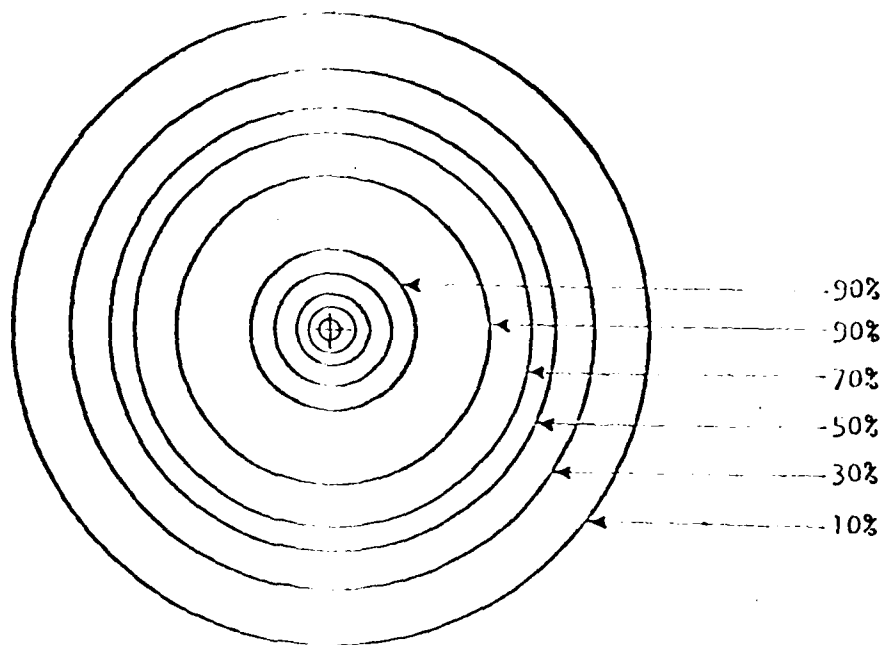


Figure B 2. Single Sensor Coverage Template. (Circles show distance from sensor location, center, at which percentage values of the peak detection rate occur.)

When several radars have overlapping coverage as shown in Figure B3, the probability of detecting your aircraft at a point within areas of overlap is higher than it would be at that same point if only one radar could detect at that point. Thus there is a joint detection capability throughout areas of overlap. The points where joint probabilities of detection are equal are connected together to form contours as shown in Figure B4. The contours have the same general meaning as the concentric circles in Figure B2, that is, each contour is the set of points where detection capability is some specified percentage of the peak joint detection capability. The set of contours is analogous to a topographical map. The difference is that each contour on a topographical map corresponds to an altitude above sea level and each detection capability contour corresponds to a detection capability between zero capability and the peak capability.

B. MEASURING THE GOODNESS OF AN AIR STRIKE PATH: THE UTILITY FUNCTION

The problem is to select an optimal air strike path, so an appropriate utility criterion function is one which measures the "goodness" of an air strike path. The two variables selected to determine the goodness of an air strike path are fuel consumption along the path and probability of being detected by one or more enemy radars prior to reaching the target. The utility criterion function incorporates a tradeoff between minimizing the probability of being detected by enemy radars on one hand and maximizing the fuel remaining upon arrival at the target on the other.

The utility function takes on any value between 0 and 100, with higher utility values corresponding to "better" paths. A family of parameterized curves from the utility function is shown in Figure B5. The figure shows that as the probability of being detected by enemy sensors decreases, the utility value goes up. Also, if the probability remains constant, the utility value increases as fuel consumption drops. It is obvious why it is desirable to minimize the probability of being detected by enemy sensors. The rationale for encouraging fuel preservation is that if detection occurs at any time up to arrival at the target, there should be as much fuel left as possible in order to do some flight maneuvering to try to return safely.

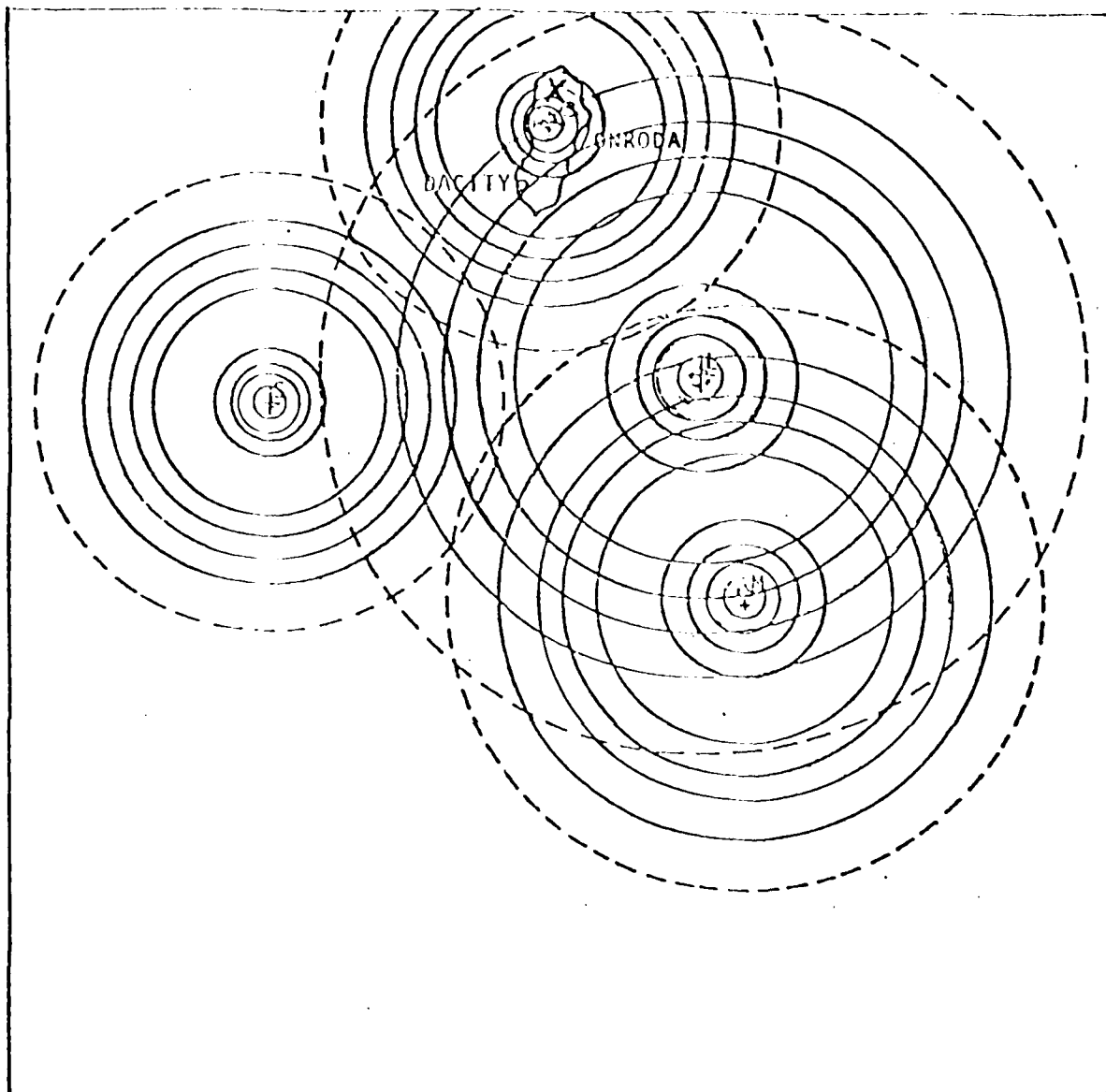


Figure B3. Display with Four Sensor Coverage Templates Shown.

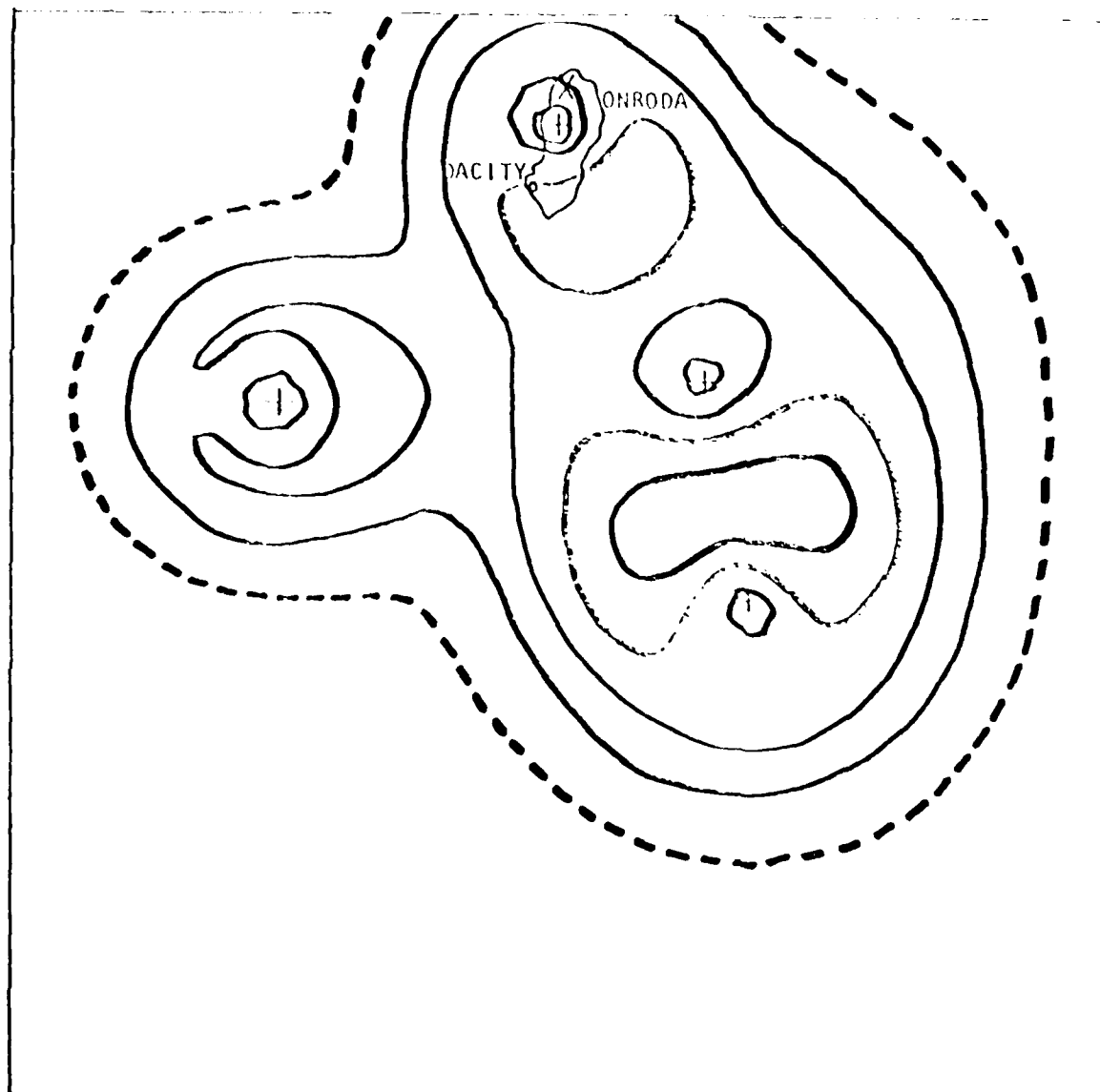


Figure B4. Contours Showing Joint Probabilities of Detection for Four Radars Located at Positions Marked with a Cross (+).

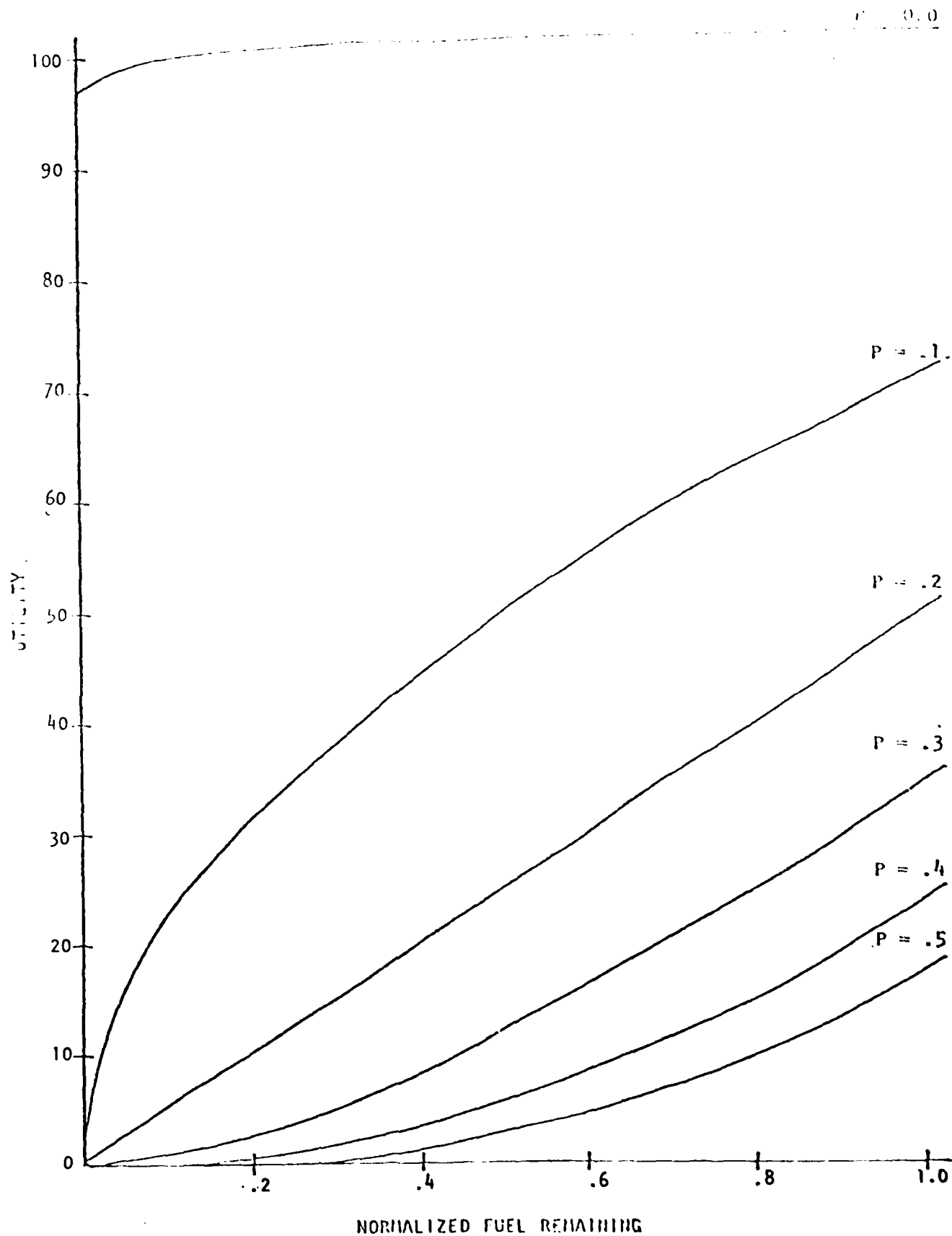


Figure B5. Family of Parameterized Utility Function Curves.

In general the two goals of minimizing fuel consumption and minimizing the probability of being detected are incompatible. A nontrivial optimal air strike path thus requires a reasonable compromise between the two goals.

C. CHARACTERISTICS OF THE AIR STRIKE PROBLEMS USED IN THE EXPERIMENT

The setup for the present experiment includes the starting "point" or first trial solution. In the air strike problem, the starting "point" is (a) five path legs connecting the air strike launch point and the target and (b) speeds for each leg. The legs are specified by picking four "way points" between the launch point and target. Speeds are selected from a range of 250 to 1000 knots. After the start point has been specified, the operator may attempt to find a better combination of way points and speeds. He or she does this by exploring changes in the location of each way point and the speed for each leg, and looking for any improvement in the path utility.

D. OPERATION OF THE COMPUTER PROGRAM

At the beginning of a problem the display will appear as shown in Figure B6. The path from launch point to target is a straight line with way points indicated at $1/5$, $2/5$, $3/5$, and $4/5$ of the straight line distance. Speed for each leg is initially set by the program as 600 knots as indicated on the plot at the left in Figure B6. The subject uses the appropriate buttons on the function button box (see Figure B7) and the joystick to change the position of the four way points. He uses the appropriate function buttons and the keyboard to change speed on any leg.

The subject's purpose is to investigate as many reasonable potential solution regions as possible in 15 minutes. As soon as the problem is shown on the display, the subject must decide what region he wants to explore first. He is to pick the region that he thinks is most likely to contain the best solution. He then changes the locations of the way points and speeds accordingly. At the beginning of the problem the three buttons designated as "Evaluate/Halt," "Change Velocity," and "Move Way Point" are lit on the box. In order to move a way point, push that button. When this is done the four buttons marked 1, 2, 3, and 4 will light. Then push the button corresponding to the point to be moved, i.e., 1, 2, 3, or 4. Way point 1 is the closest to

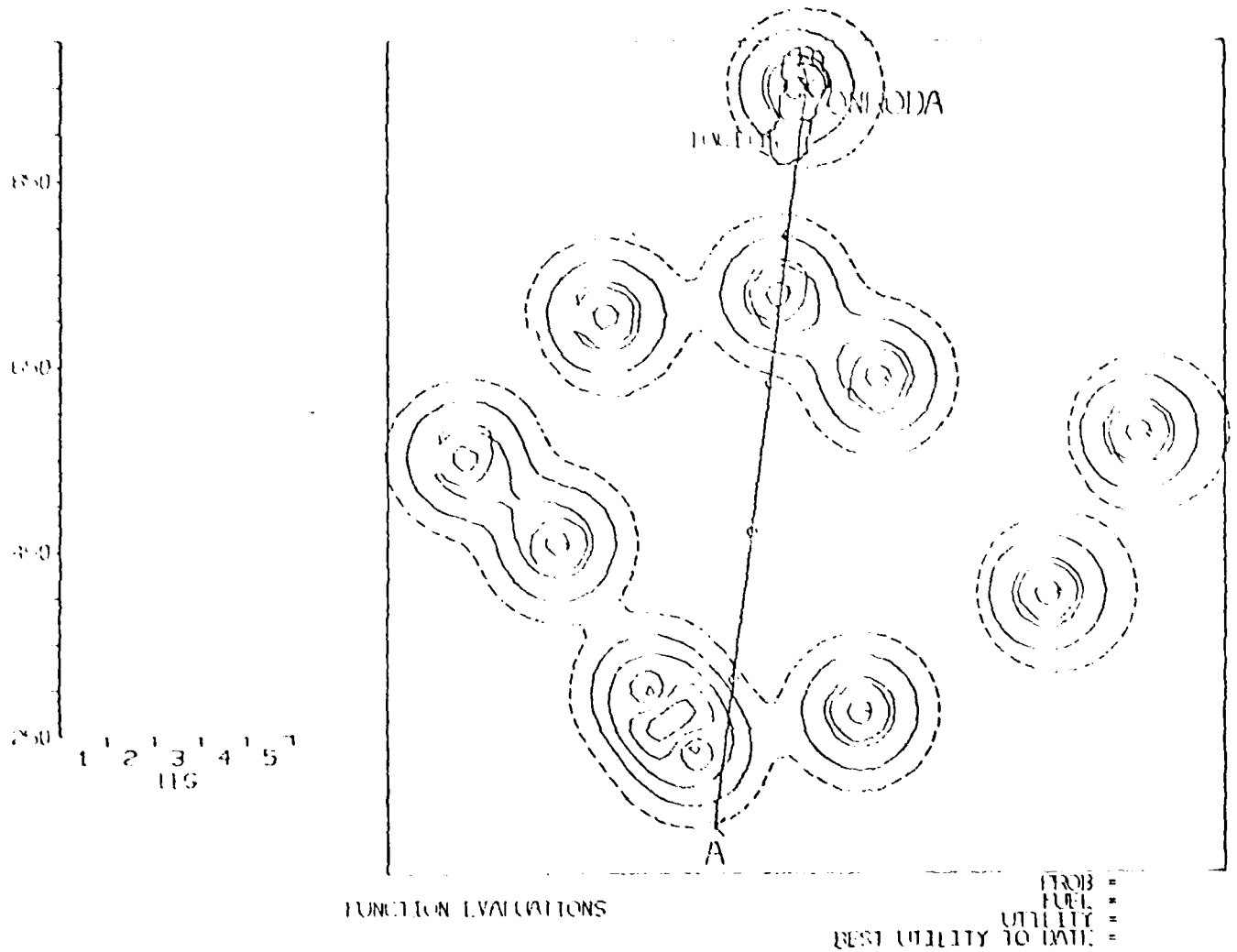


Figure B6. Display Appearance at Beginning of Problem to be Solved.

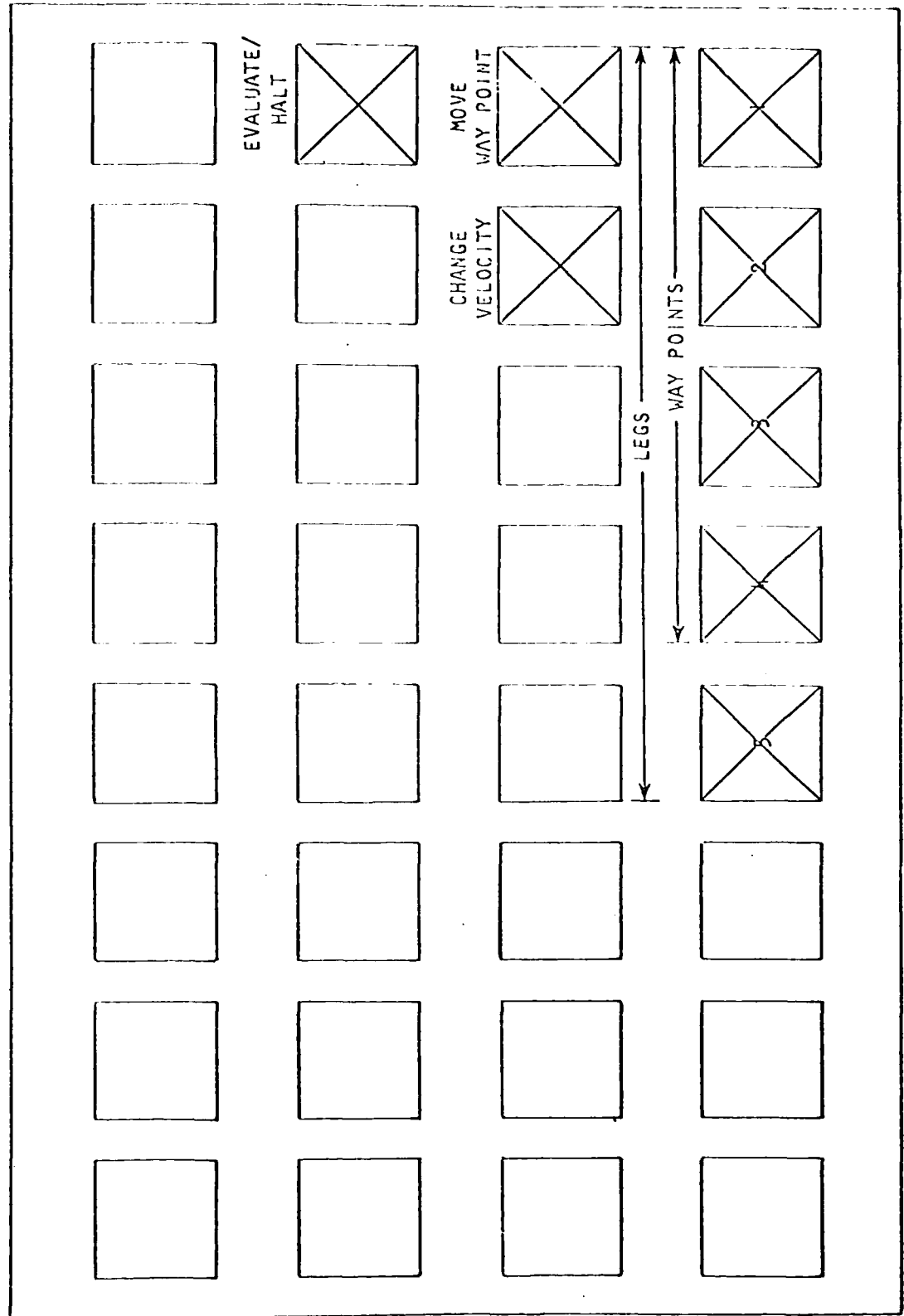


Figure 87. Function Button Configuration.

the beginning of the strike path and 4 is the nearest to the end (ONRODA Island). Moving the way point is accomplished with the joystick. When a single way point is changed, a second way point can be changed by pressing "Move Way Point" and the appropriate number of the way point. The act of pressing "Move Way Point" records the position of the last way point that was changed.

To change a speed on one of the five legs, push "Change Velocity." The five buttons marked 1, 2, 3, 4, and 5 will light. Leg 1 refers to the leg closest to the path start point and leg 5 refers to the path closest to the end point (ONRODA Island). Then push the button corresponding to the leg for which you want to change speed and:

1. Use the teletype keyboard to input the speed you want used on the selected leg. Put a decimal point at the end of the number. (This is essential.)
2. Push the teletype key marked "CR."

Thur, if you wanted to change the speed on leg 3 to 850 knots, you would:

1. Press function button "Change Velocity"
2. Press function button "3"
3. Press teletype key "8"
4. Press teletype key "5"
5. Press teletype key "0"
6. Press teletype key "."
7. Press teletype key "CR"

When you have changed all the way points and speeds to those you want, then press the function button marked "Evaluate/Halt." The program will then evaluate your starting point consisting of the four way points and five speeds. The values for probability of detection, fuel consumed, and path utility will be displayed in the lower right corner of the screen (refer back to Figure B6). Note that the number of paths tried is recorded as "function evaluations" in the lower left corner.

As a guide to optimum use of problem time, the operator should count the number of regions that could reasonably be expected to contain the best path. Dividing 15 minutes by the number of regions to be explored indicates

approximately the number of minutes the operator should devote to each region. Depending on the problem, there will be enough time to explore 3, 4, or 5 regions.

At the end of 15 minutes the computer will have stored:

1. The utility of the path comprised of the first way points and leg speeds entered by the operator.
2. The utility of each best-solution-to-date at the end of each minute, excluding the first minute.

These data are the data that will be used in the analysis of operator generated data.

E. GUIDELINES

There are two types of data being analyzed:

1. Utility of the path comprised of the first way points and leg speeds entered by the operator. Thus the operator's first goal is to do the best he or she can on this.

2. Operator performance will be calculated at the end of each trial by adding the 14 utilities of the best-solution-to-date at the end of each minute (excluding the first minute) and dividing this sum by 14. Thus, operator performance for the entire trial is the average of the 14 utilities. The operator's second goal is to maximize this average. In general this is done by exploring the regions which could contain the best path in the order of the estimated likelihood that each contains the best path. This is compatible with the operator's first goal because, if the operator is correct concerning the region which contains the best path, then the average utility will be nearly equal to the utility of the best path. This is true because the computer only stores the best utility to date and will therefore not store the utilities of paths investigated after the first if the first region explored contains the best path.

Other general rules to be used on the problem are:

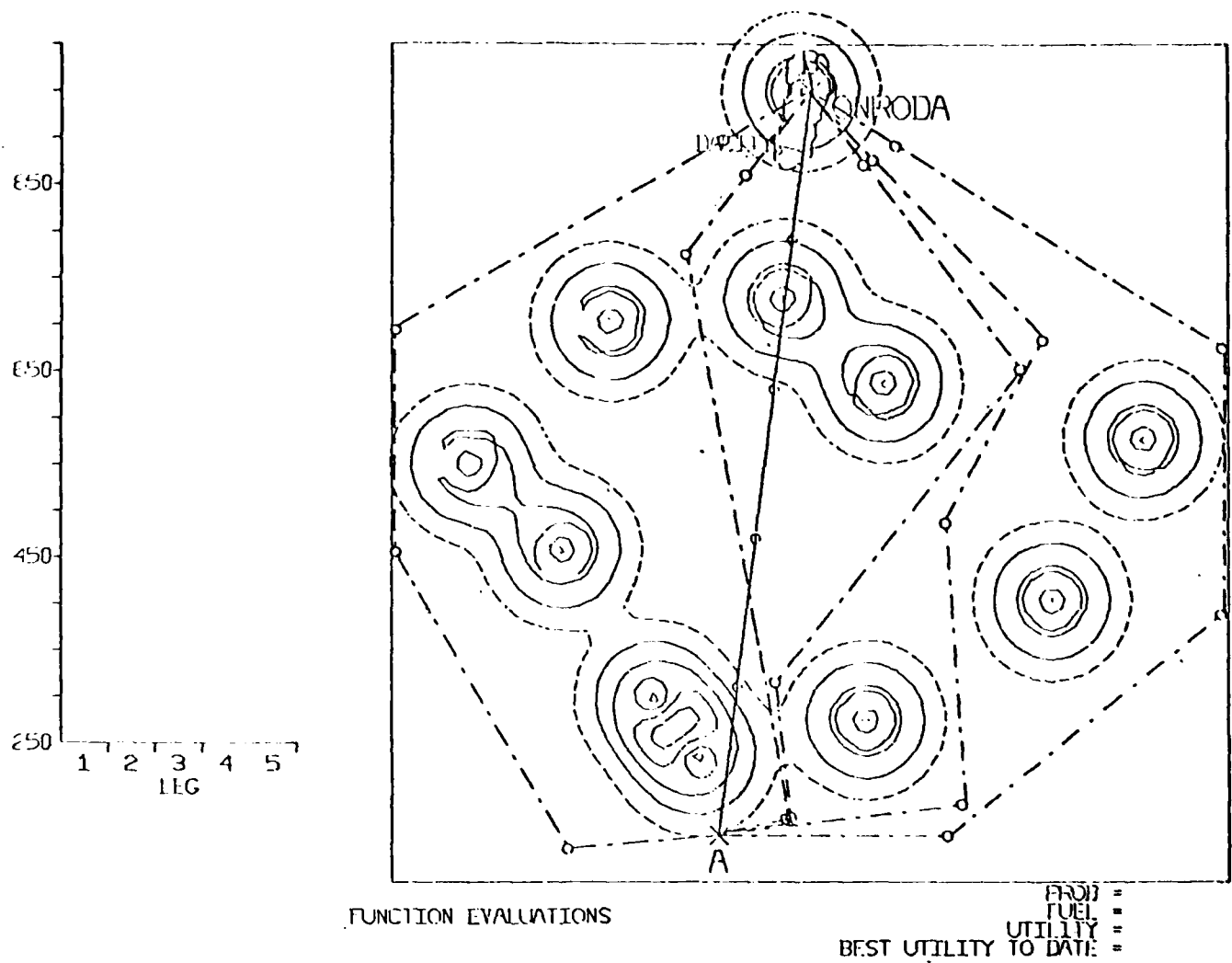
1. Those portions of a path that are completely outside the detection contour should be transited at low speeds.

2. Those portions of a path that traverse a high detection probability contour should be transited at high speeds. In particular, the last leg of the path to the target should be transited at high speed since it must pass through the high detection region around ONRODA airport. It is best to locate the last way point just outside this region and use a speed such as 999.

3. Paths should be drawn to pass through low detection probability regions. However, a completely roundabout path that avoids detection contours completely is not a sure winner because long paths use a lot of fuel.

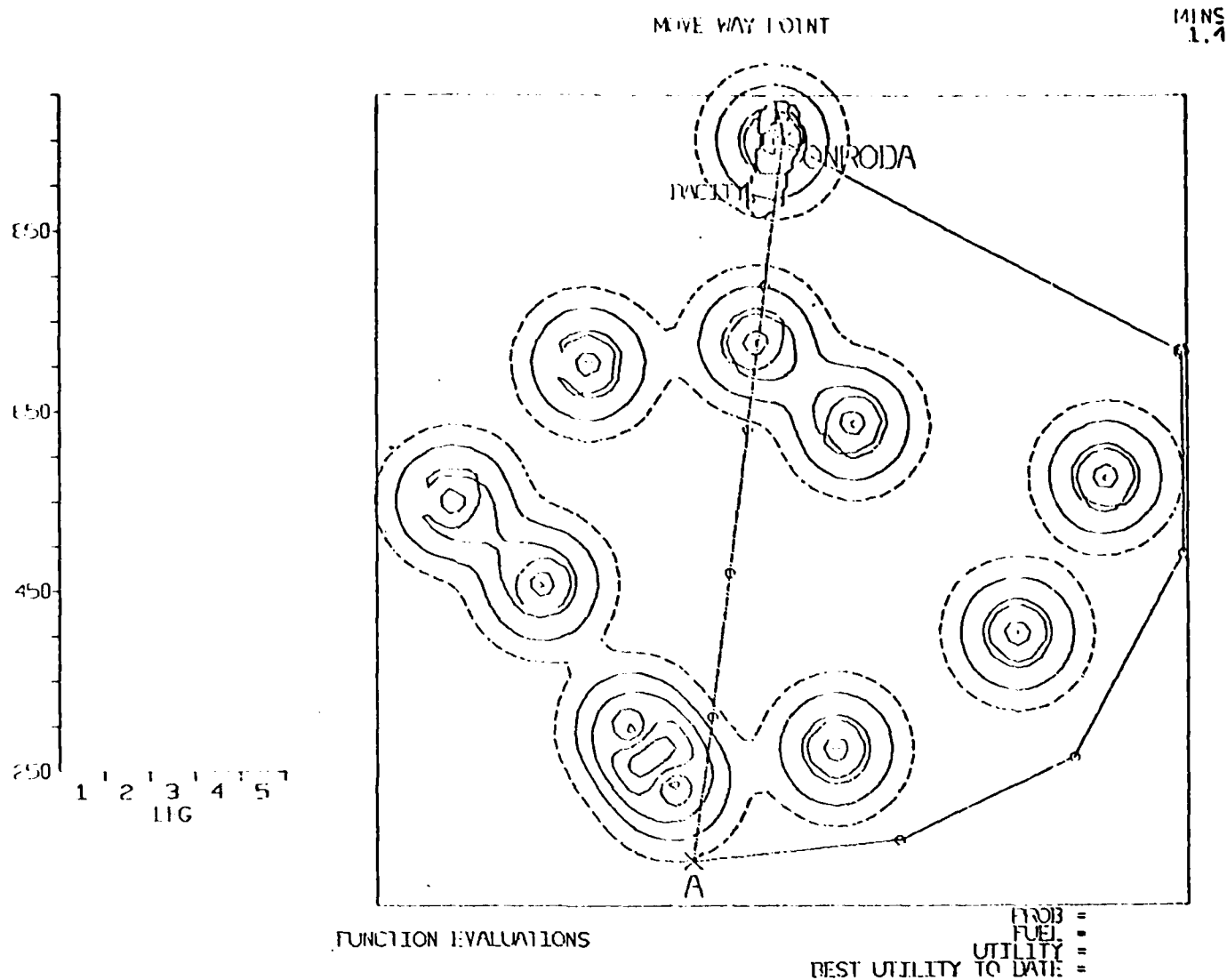
4. When crossing detection regions, it is a good idea to place way points on both sides of the region, just outside the lowest detection probability contour.

The following nineteen plates illustrate these points using a sample scenario. Note that the speed/leg graph shown on the left of each plate is from an earlier version of the program; it has been replaced by a simple table of speeds. These plates were generated using the optimizer. For now, all you need to know about the optimizer is that it only finds local, not global, best solutions. In other words, it only explores a region around the operator-defined initial solution.



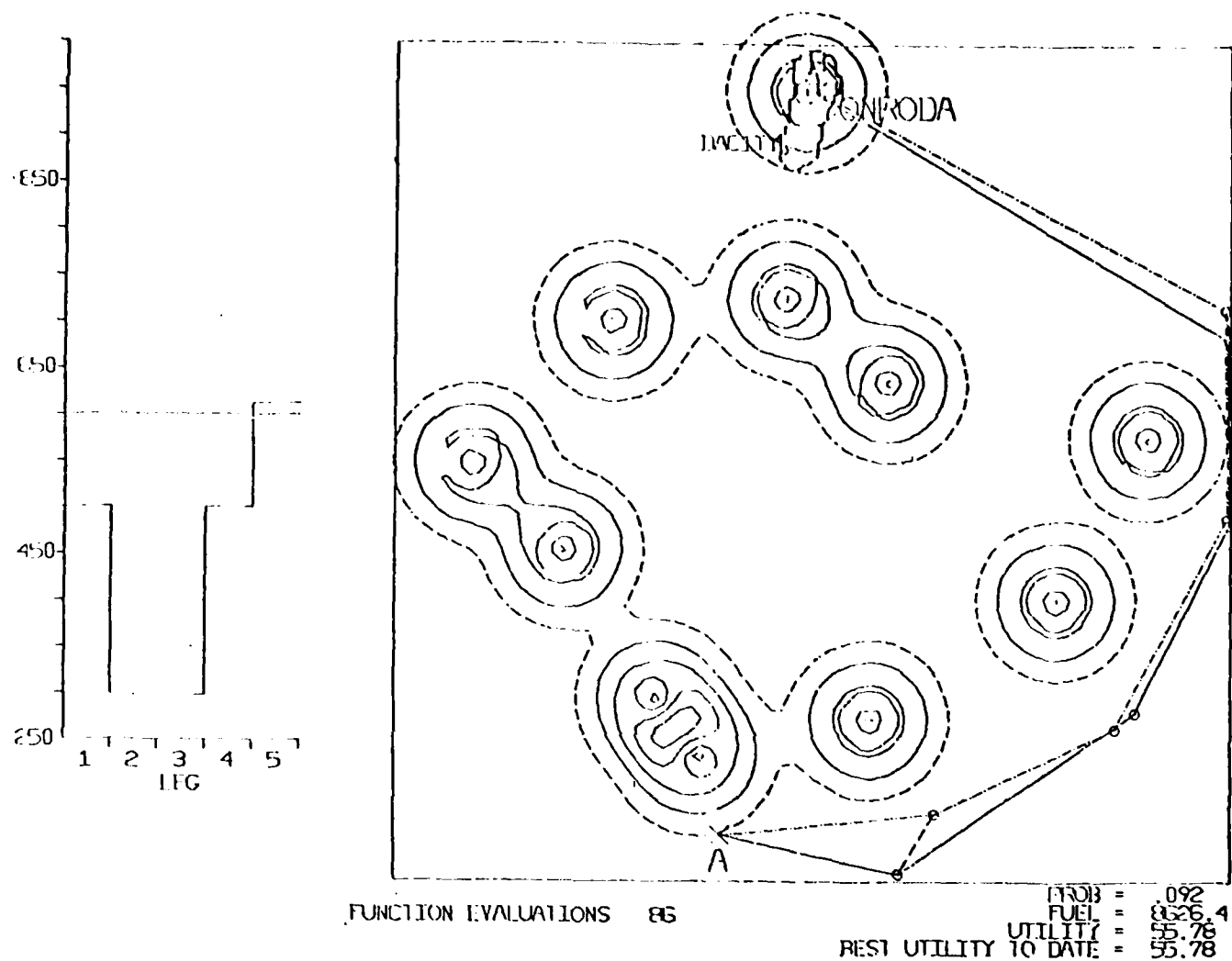
This is how the display appears at the beginning of the problem. Five potential best paths are shown as dot-dash-dot lines.

Figure B8. First Plate, Example Problem.



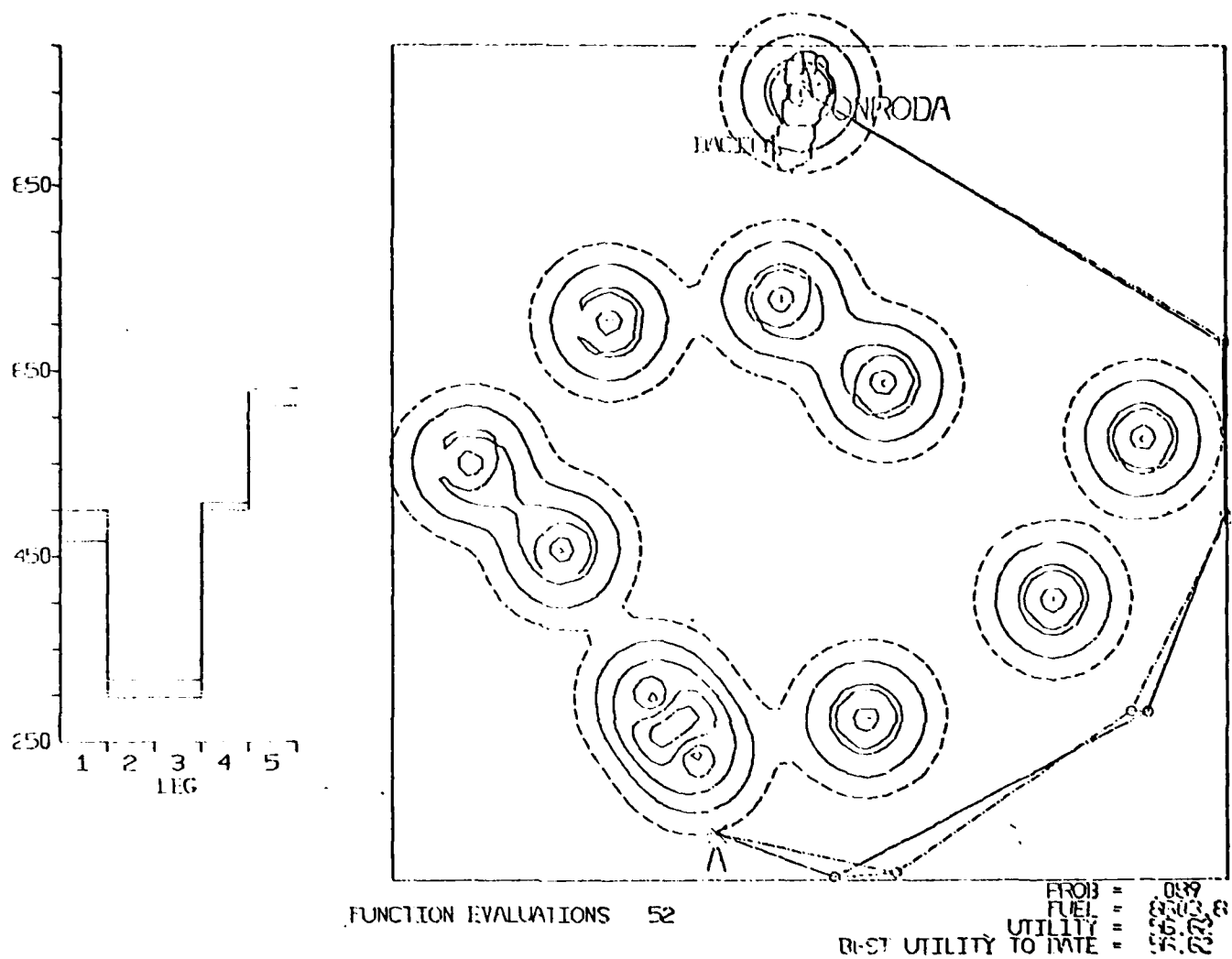
The operator chose to explore paths from right to left. It would have been better to have configured the path so that the last leg began just outside the contours around ONRODA. The previous starting path remains on the display as a dot-dash-dot line.

Figure B9. Second Plate, Example Problem.



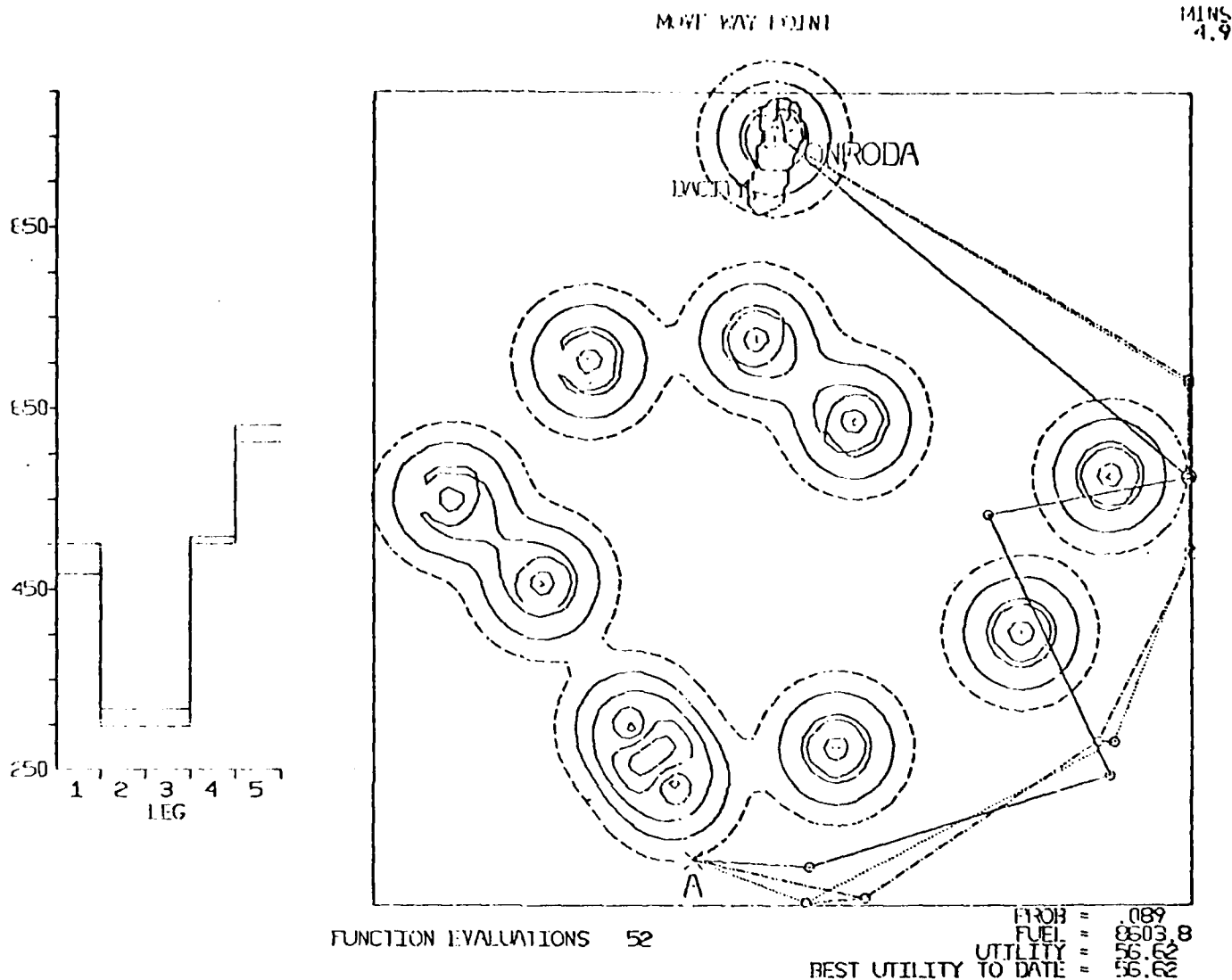
The operator stopped the algorithm at the end of 86 evaluations in order to get this picture. Note that the solution moved the first way point down in order to get away from the contours above the point.

Figure B10. Third Plate, Example Problem.



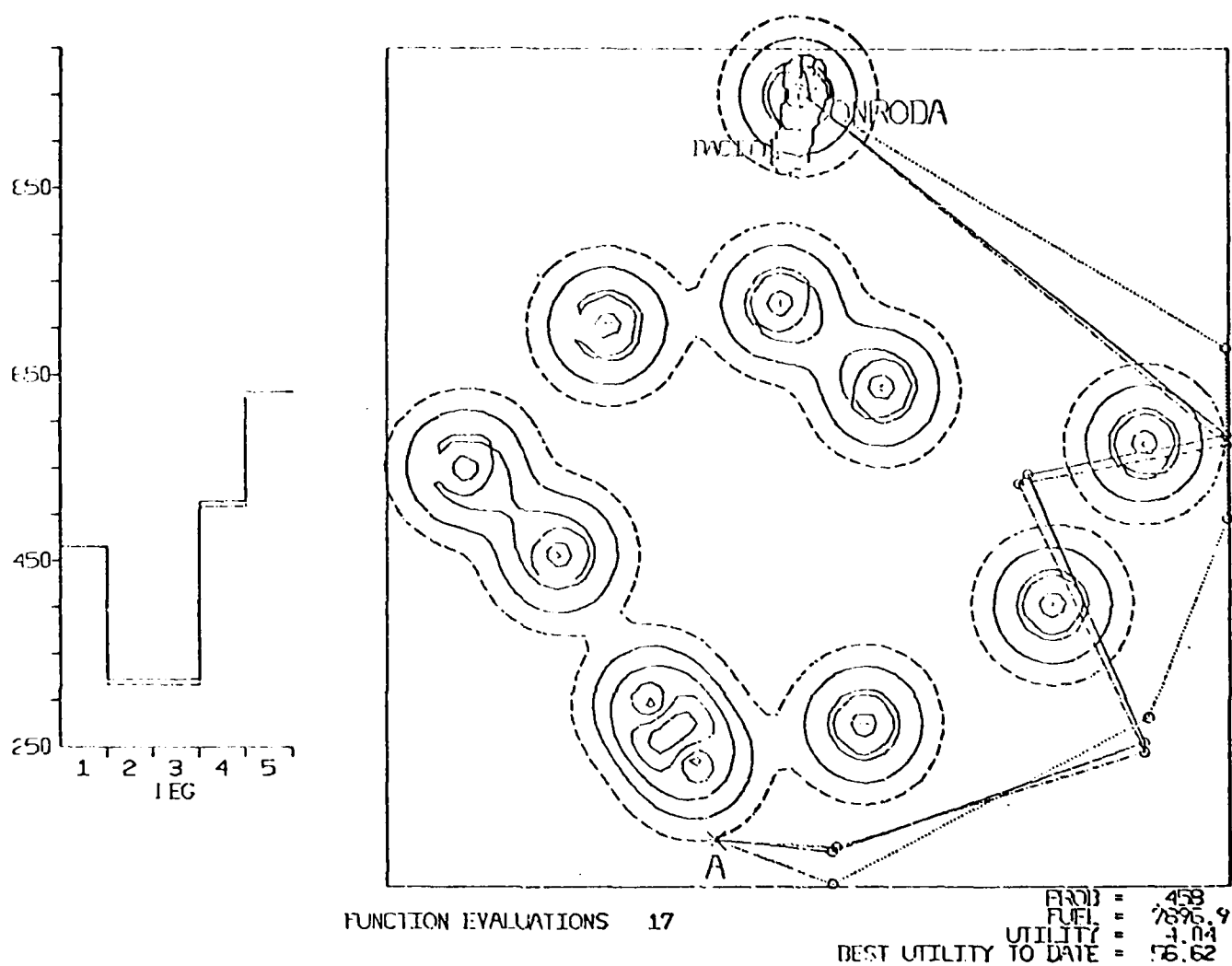
The operator restarts the algorithm without making any changes. At the end of another 52 evaluations (138 total), the operator stops the algorithm because (a) the step sizes being considered are very small and therefore the possible utility improvements will also be small, and (b) the utility hasn't increased very much in the last 25 or so evaluations.

Figure B11. Fourth Plate, Example Problem.



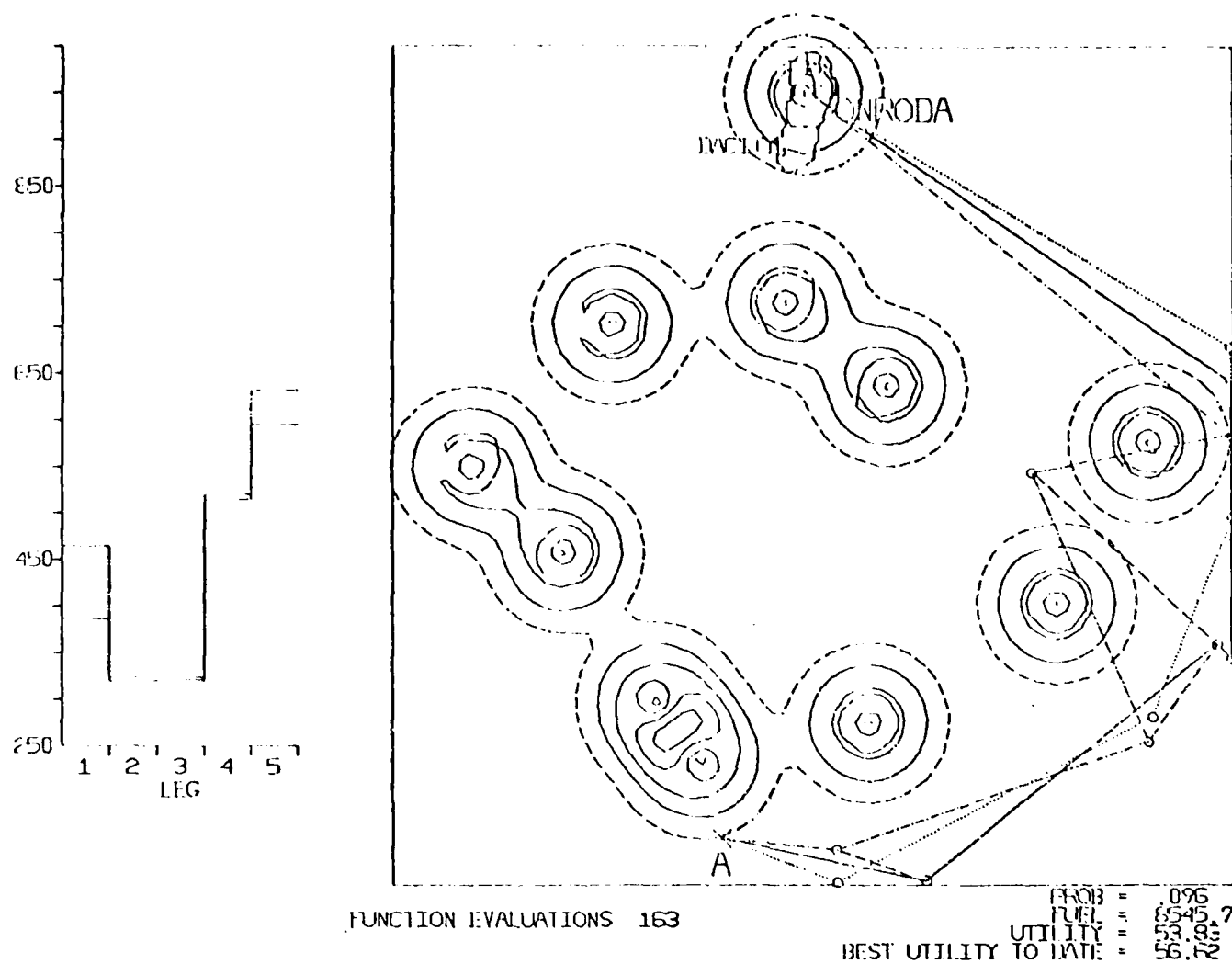
The operator puts the third and fourth way points in an illogical combination of places and makes small adjustments to the other two way points. The point will be to see what the algorithm does.

Figure B12. Fifth Plate, Example Problem.



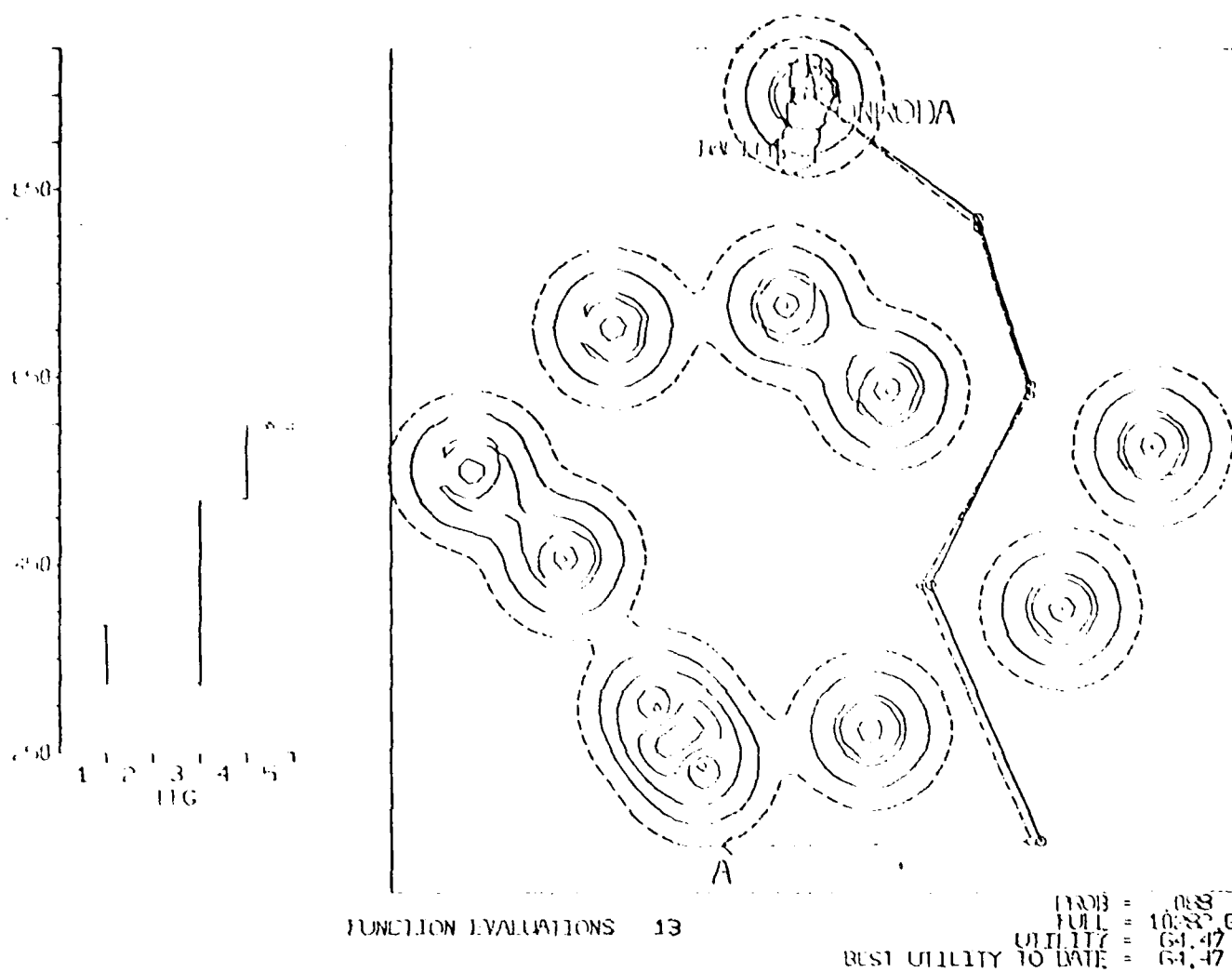
At the end of only 17 evaluations not much has happened.

Figure B13. Sixth Plate, Example Problem.



At the end of 163 evaluations the algorithm has found its way over to a much better position for the third way point but the utility is not as good at 163 evaluations (53.83) as it was at 140 evaluations with the earlier, better selection of way points (56.62 for the starting path of Figure B10).

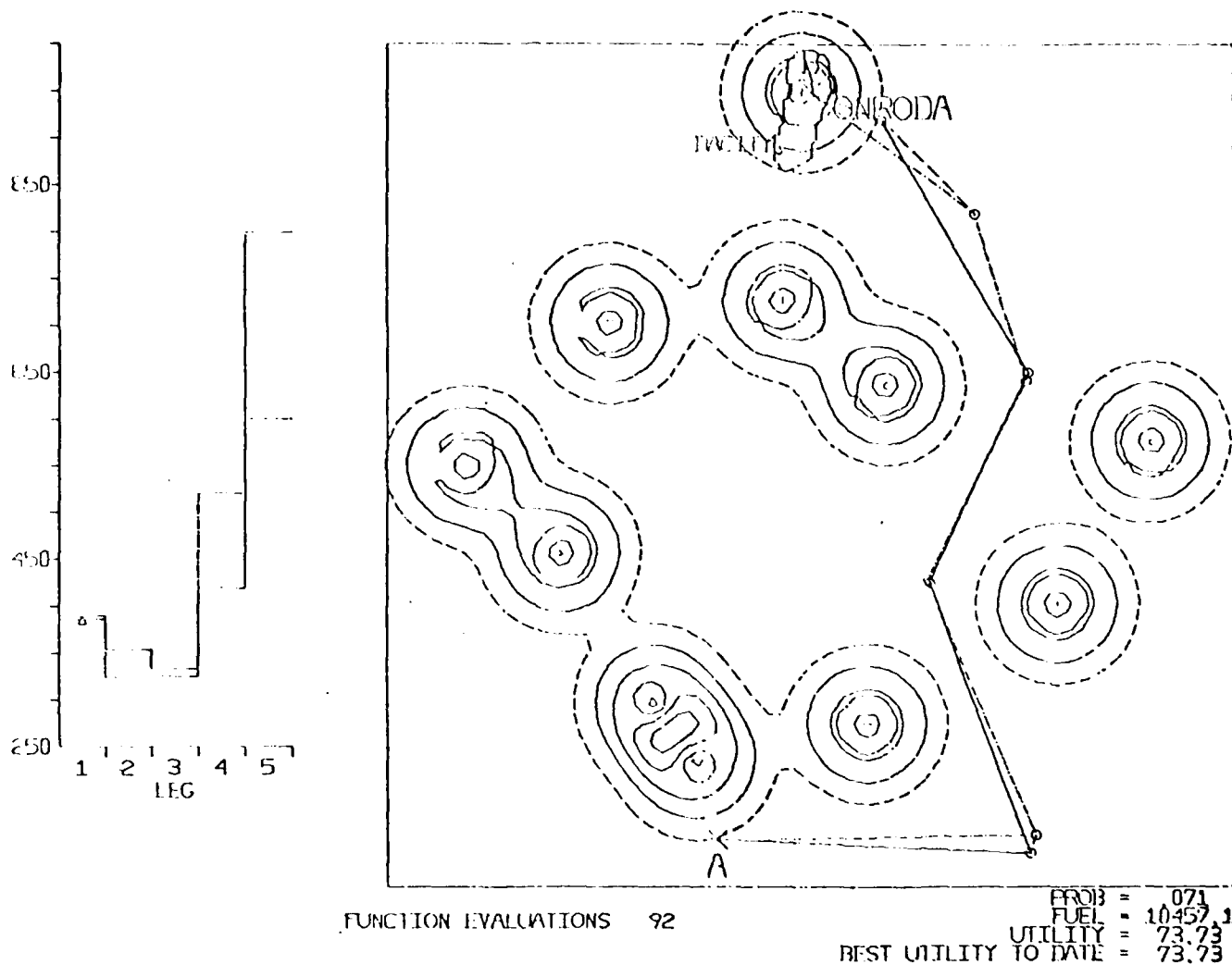
Figure B14. Seventh Plate, Example Problem.



The operator selects a new set of way points and the algorithm begins to explore around these. Again, he should have placed the last way point closer to ONRODA.

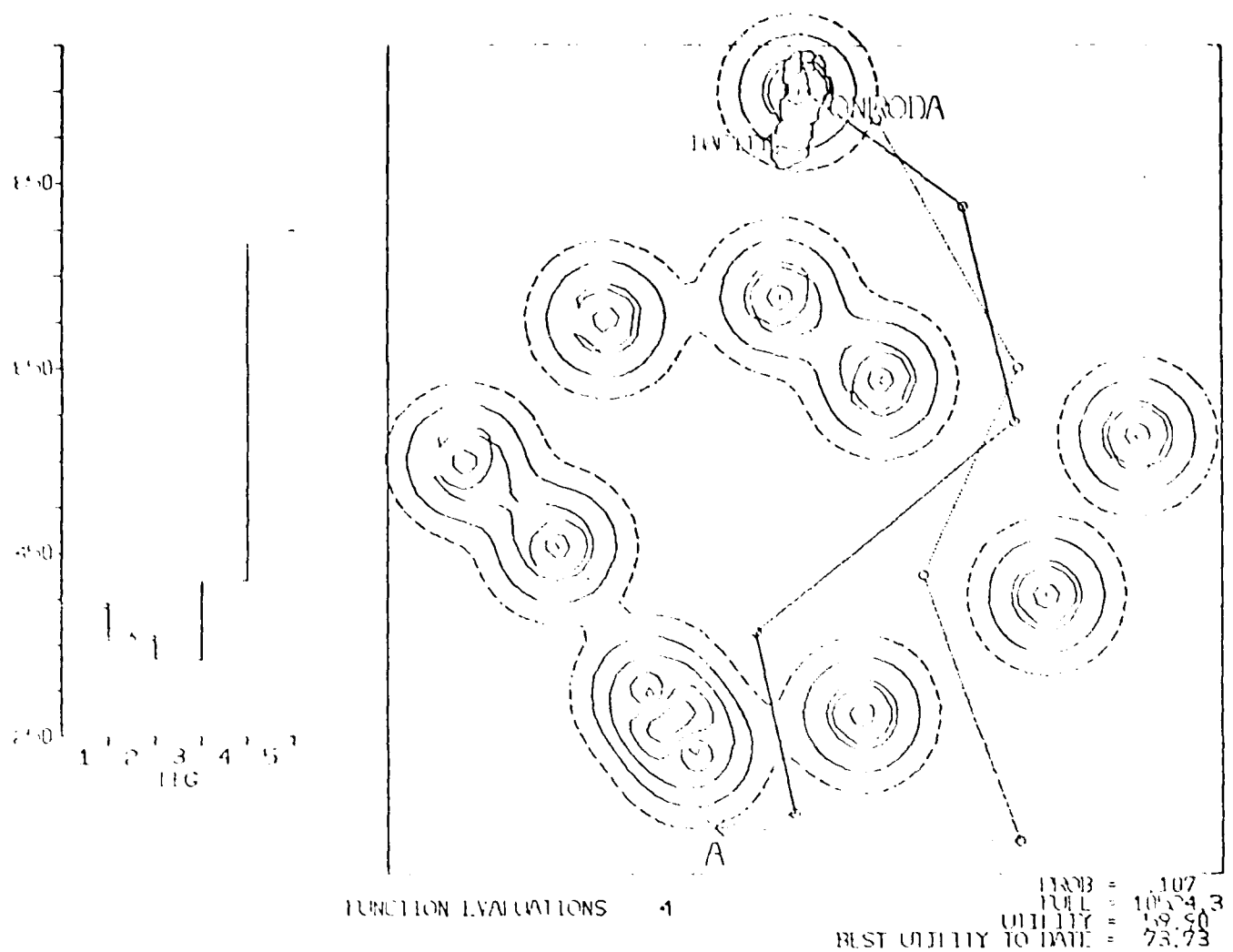
Figure B15. Eighth Plate, Example Problem.

MINS
10.0



At the end of 92 evaluations the operator stops the algorithm. Note that the algorithm has moved the last way point much closer to ONRODA and has greatly increased the speed for the last leg.

Figure B16. Ninth Plate, Example Problem.



The operator has already selected way points for the third path to be explored by the algorithm. Utility is 59.90 at the end of four evaluations, and then the operator stops the algorithm. He has decided to change the speed on a particular leg and accordingly pushed the "Change Speed" function button. The prompt "Choose leg" then appears at the top of the display. Then he pushes the function button corresponding to the desired leg.

Figure B17. Tenth Plate, Example Problem.

AD-A087 668

INTEGRATED SCIENCES CORP SANTA MONICA CALIF
COMPARISON OF OPERATOR AIDED OPTIMIZATION WITH ITERATIVE MANUAL--ETC(U)
JUL 80 M D SCHECHTERMAN, D H WALSH
ISC-215-6

F/G 5/8

N00014-75-C-0811

NL

UNCLASSIFIED

2-12

20
N-11, 68

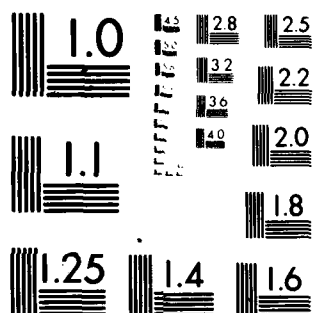
END

DATE

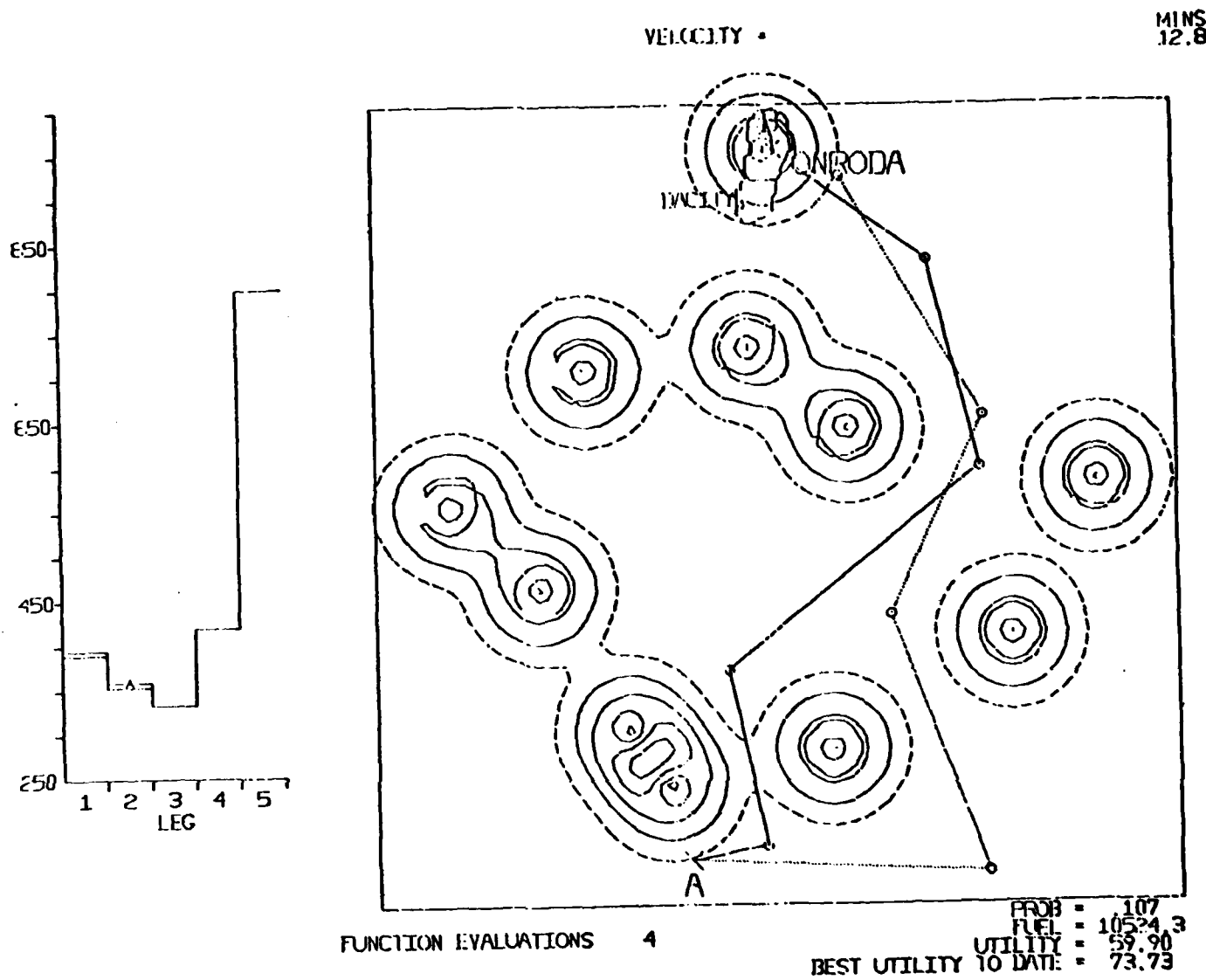
FILMED

9 80

DTIC

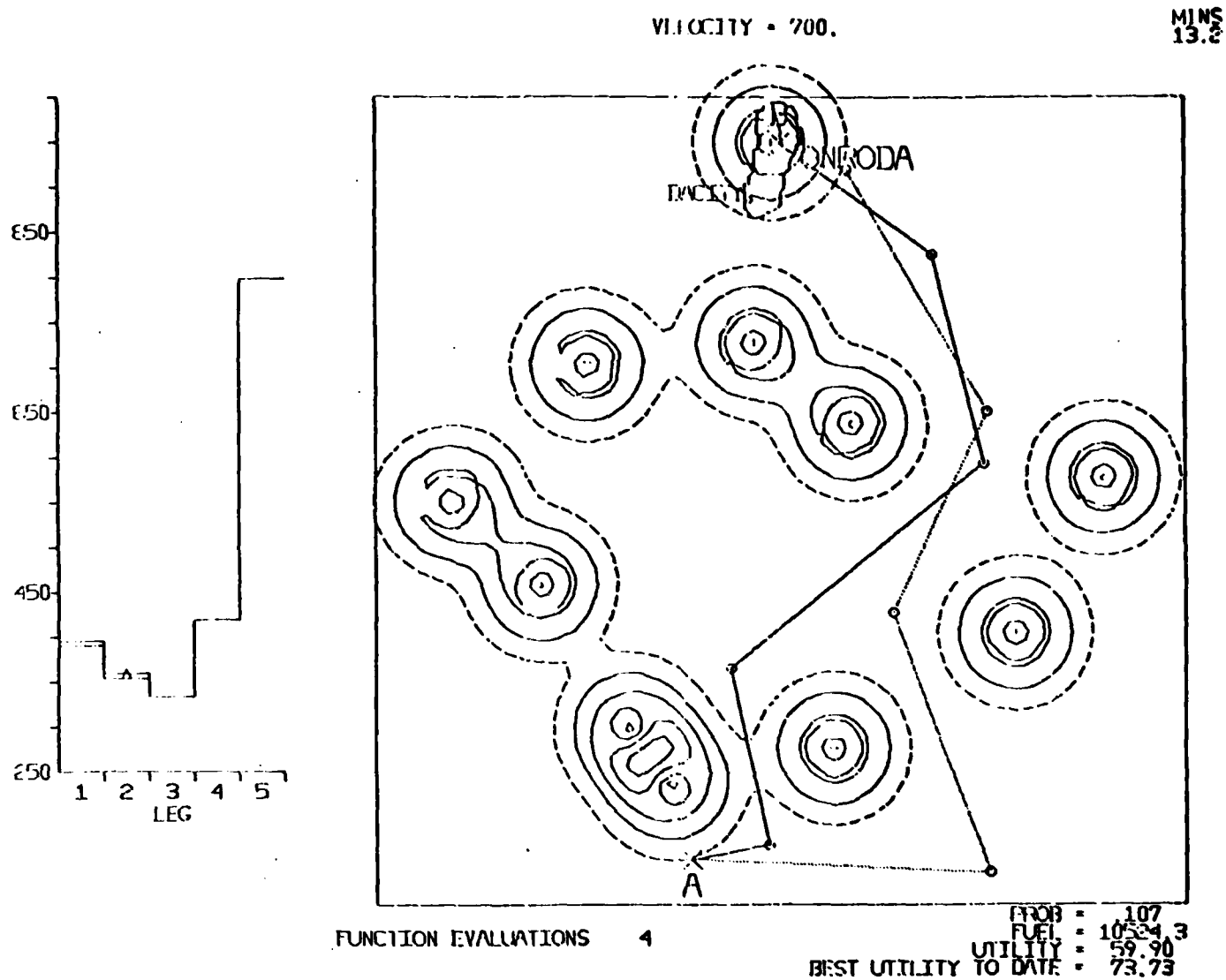


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



Immediately the prompt "Velocity = " appears at the top of the display.

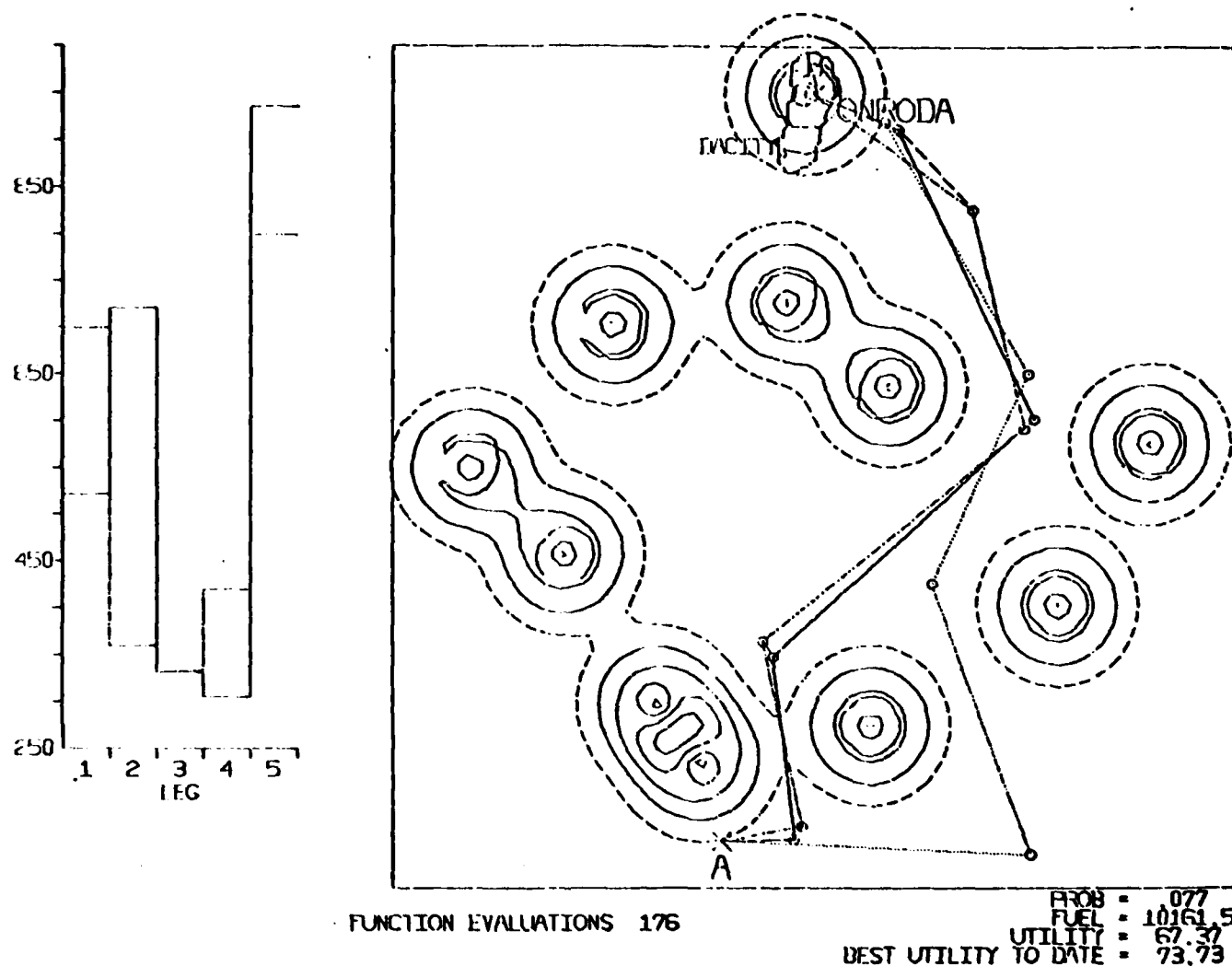
Figure B18. Eleventh Plate, Example Problem.



The operator then types "700. CR " and 700 appears at the top of the display. The operator restarts the algorithm.

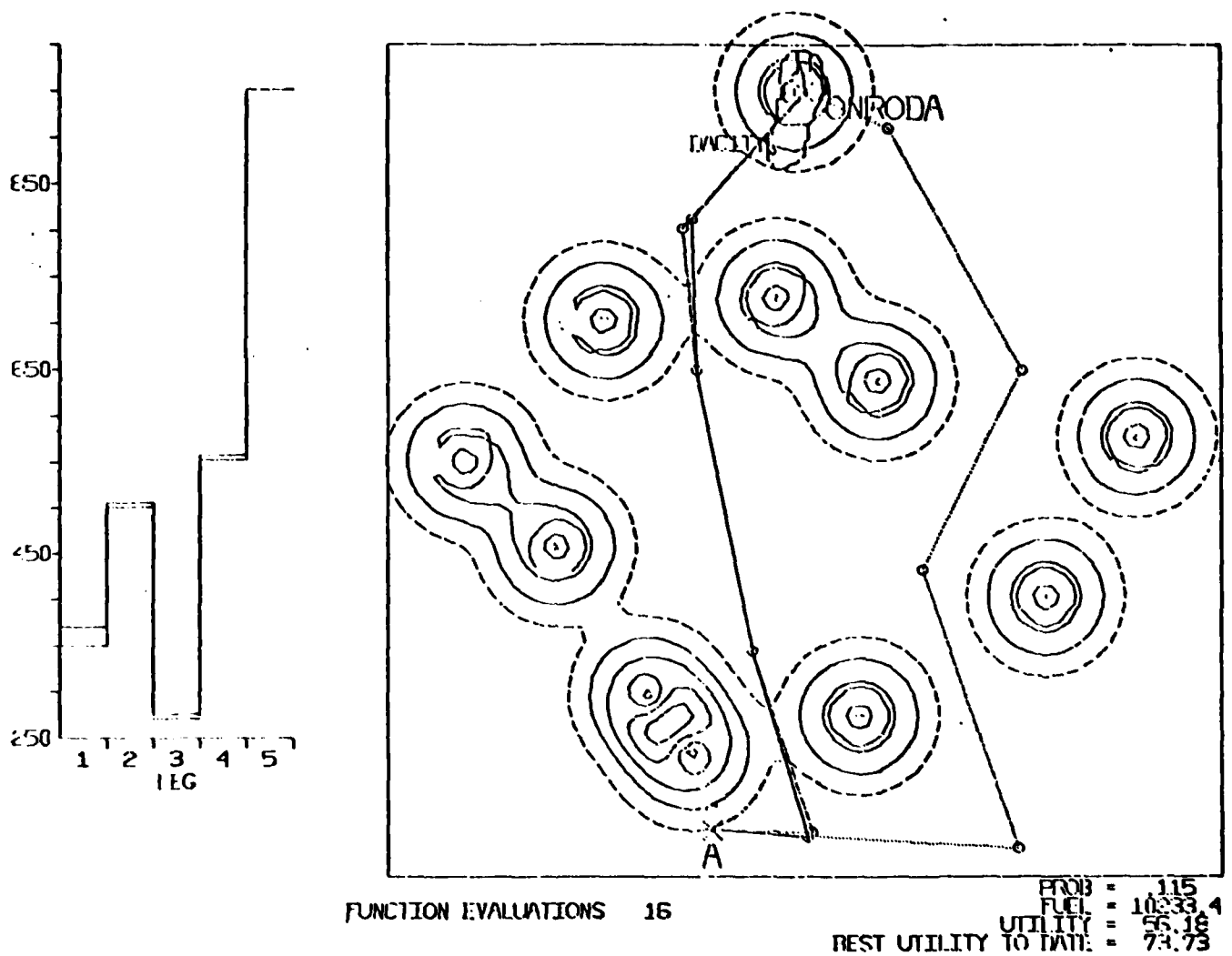
Figure B19. Twelfth Plate, Example Problem.

MINS
16.3



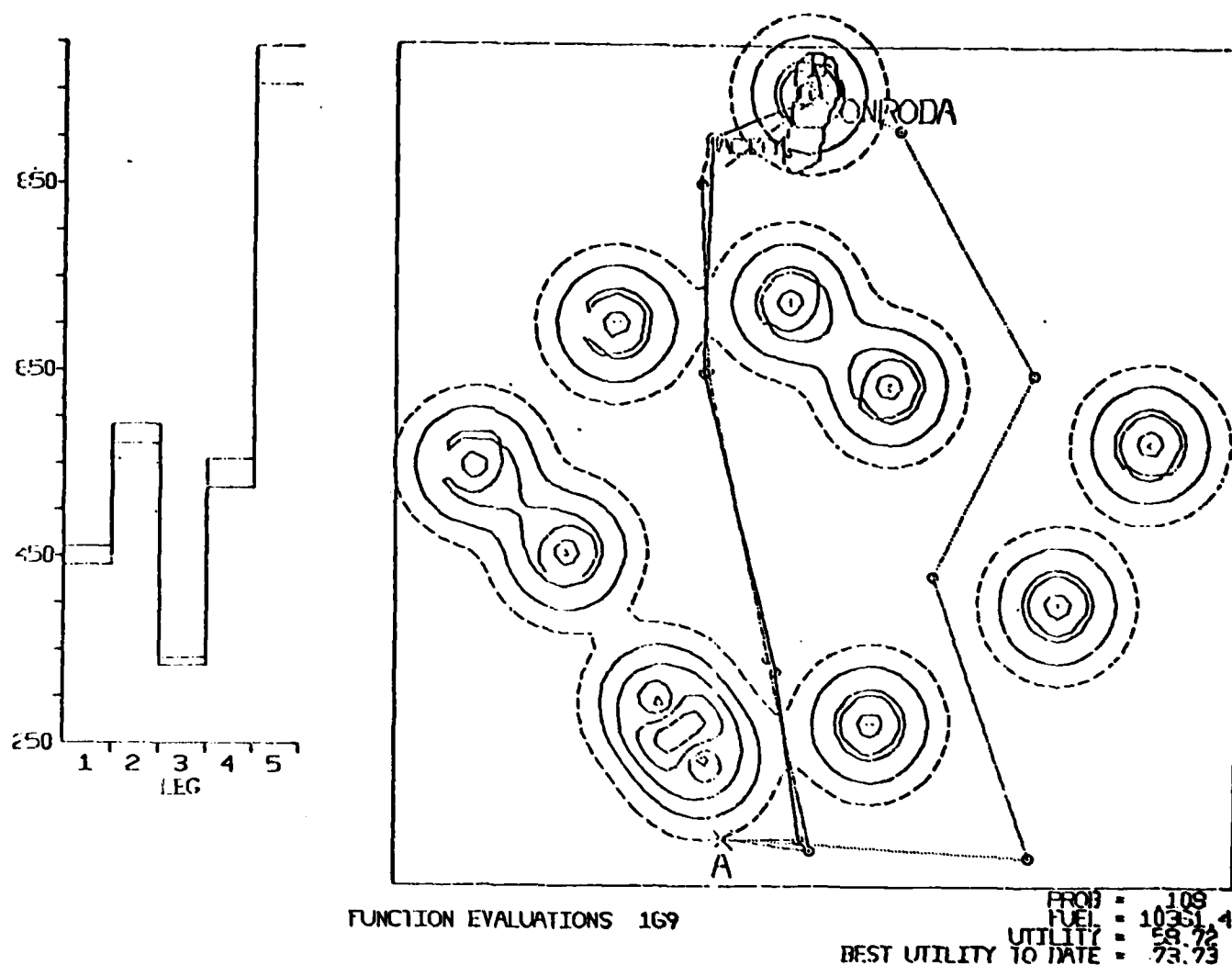
The operator stops the algorithm at the end of 176 evaluations. Note again that the algorithm has moved the last way point much closer to ONRODA. (Disregard time shown under "MINS" from this figure on.)

Figure B20. Thirteenth Plate, Example Problem.



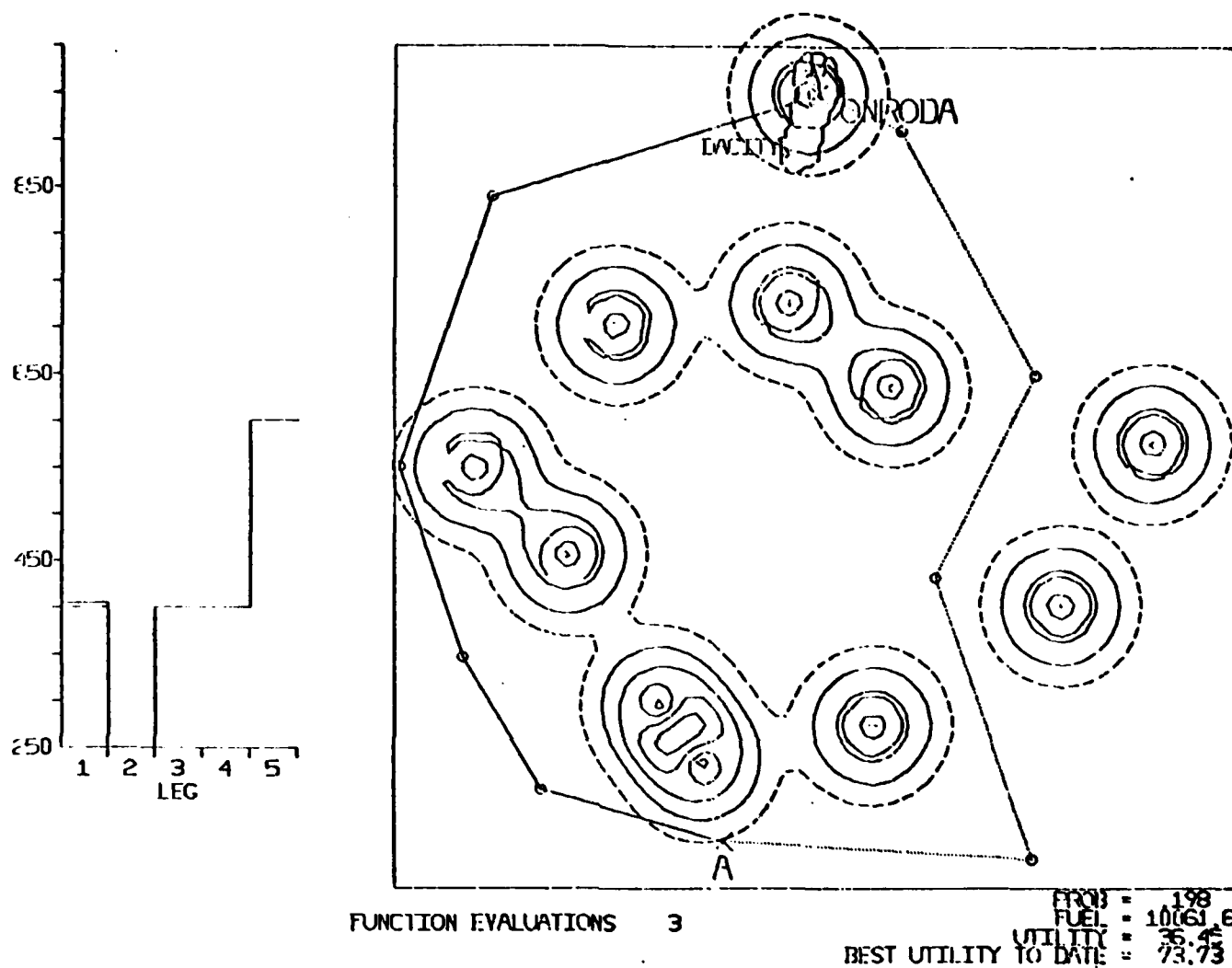
The operator has selected way points for exploring the fourth path and started the algorithm. At the end of 16 evaluations the utility is 56.18.

Figure B 21. Fourteenth Plate, Example Problem.



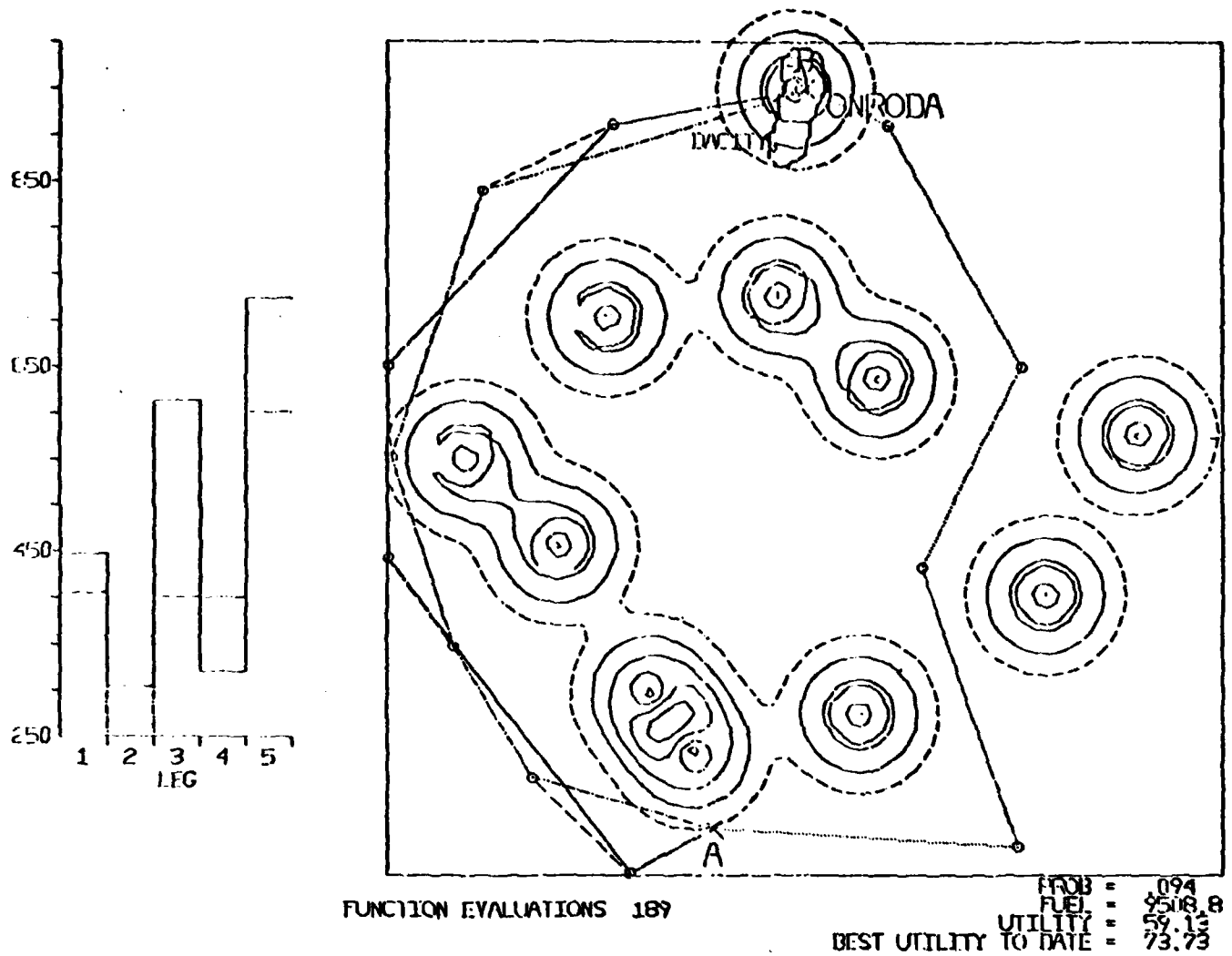
The operator stops the algorithm after 169 evaluations. Again, note that the algorithm moved the last way point closer to ONRODA. Utility is competitive with the utility for the first path explored (58.72 versus 56.62) but is significantly lower than the utilities achieved for the second and third paths explored (58.72 versus 73.73 and 67.37).

Figure B22. Fifteenth Plate, Example Problem.



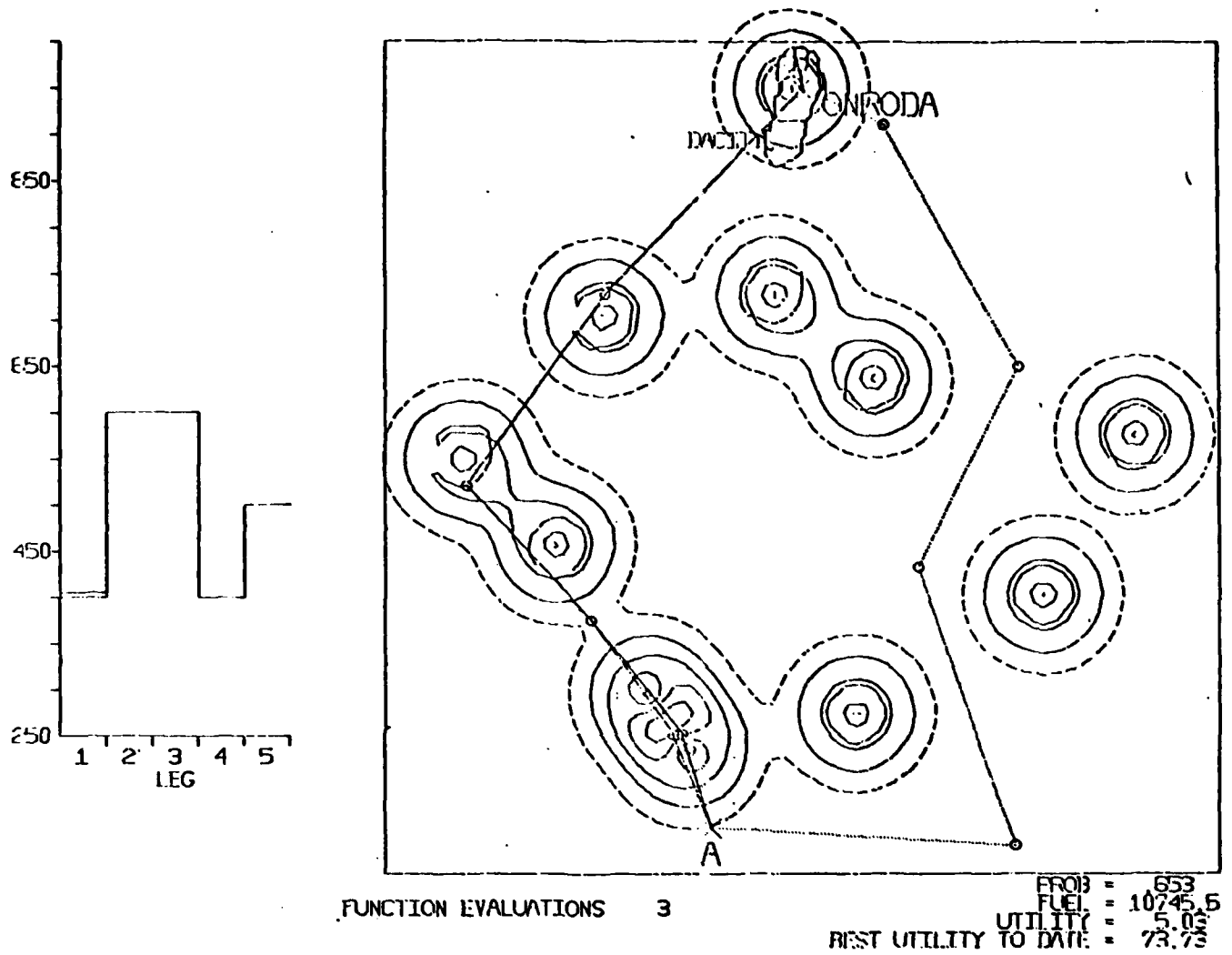
The operator resets the way points to explore the fifth path. At the end of three evaluations the utility is 36.45.

Figure B23. Sixteenth Plate, Example Problem.



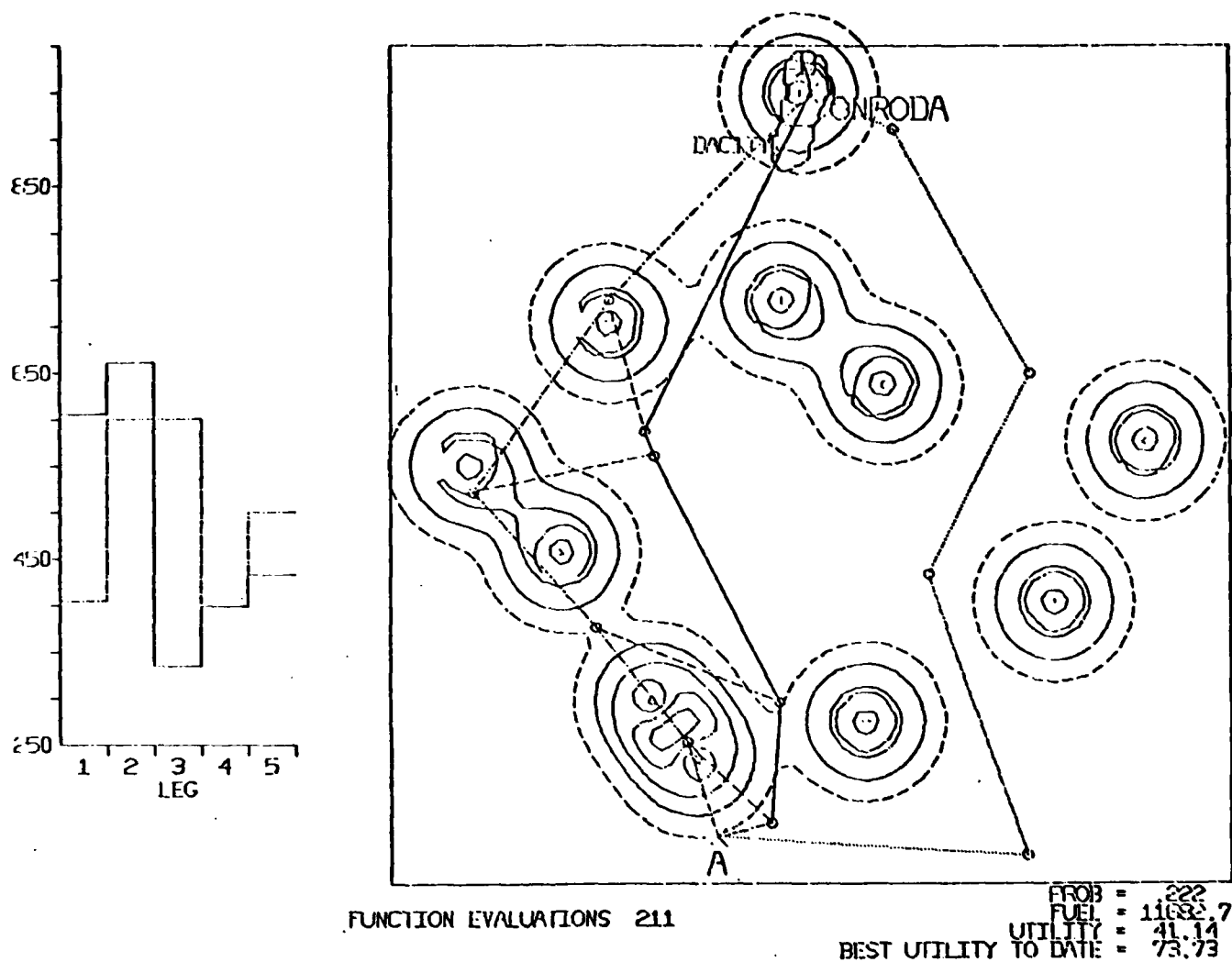
The operator stops the algorithm at the end of 189 evaluations. Note that the algorithm moved the last way point closer to ONRODA. Also, note that the first way point was moved down to get away from the contours above the starting point.

Figure B24. Seventeenth Plate, Example Problem.



The operator chooses a very poor set of way points going through high detection capability contours.

Figure B.25. Eighteenth Plate, Example Problem.



At the end of 211 evaluations the algorithm found its way over to the vicinity of the fourth path evaluated. But, clearly, it would never have found its way to the best path found by the operator interacting with the algorithm.

Figure B26. Nineteenth Plate, Example Problem.

APPENDIX C:
TRAINING MATERIALS FOR OAO

I. INTRODUCTION

Integrated Sciences Corporation is conducting a study for the Office of Naval Research that investigates ways to allocate functions between humans and computers so that their respective strengths are best used. The portion of the study in which you are participating seeks to determine to what extent, if any, selected optimization techniques can aid and thus improve the performance of a human operator when applied to a naval tactical decision problem. We call this "operator-aided optimization," or OAO for short. The optimization technique you will be working with is called nonlinear programming. Don't worry if you are unfamiliar with this technique. Even if you have never heard of it, you will learn enough about its characteristics during the training phase to enable you to perform well on the experiment.

Your role in the experiment is to act as the member of a Naval Task Force Commander's staff who is planning a tactical airstrike against the airfield on a place called ONRODA Island. Your Naval Task Force consists of aircraft carriers, their squadrons of aircraft, and escort ships. They are located approximately at the point marked with an X in Figure C1. About ten enemy ships are located in a region between your Task Force and ONRODA. Important parts of air strike planning are (a) deciding the path that your aircraft will take to get to the target and (b) strike aircraft speeds along the legs of the path. As air strike planner, you must be concerned about these two factors:

1. The probability that your aircraft will be detected before they reach the target. If they are detected before reaching the target, the enemy will be at maximum readiness to repel the air strike there. The enemy ships between your Task Force and ONRODA Island have radar that could detect your aircraft. However, the enemy ships themselves have no interceptor aircraft nor do they have guns or missiles that would be effective against your aircraft.

2. Amount of fuel left aboard your aircraft when they reach the target. It is desirable to maximize the fuel left in order to engage or avoid enemy interceptor aircraft over the target or to attack secondary targets once the primary target, ONRODA airfield, has been destroyed. Your

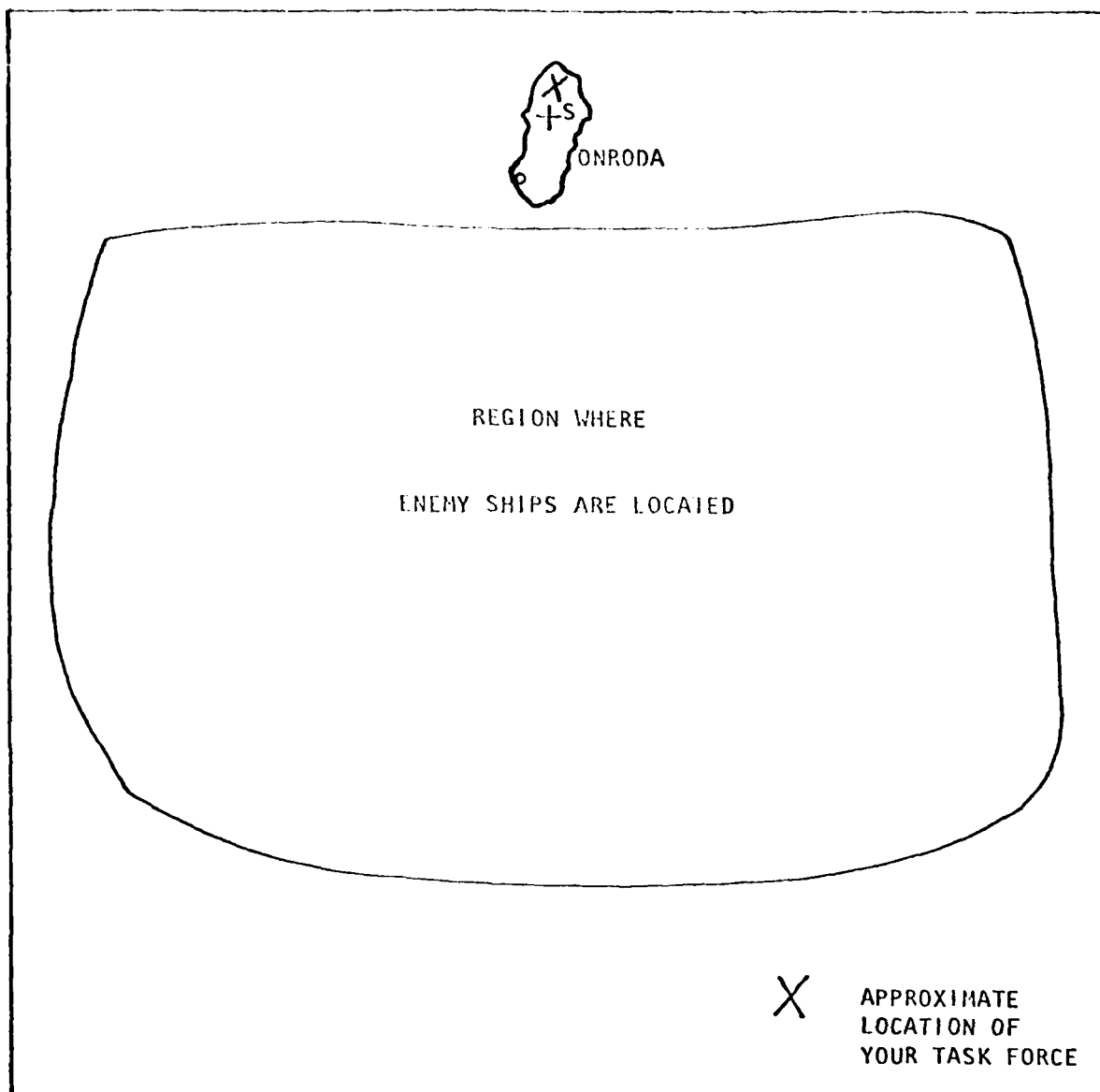


Figure C1. Map of Area of Interest.

job is to help the computer come up with the best airstrike plan between the task force and the target, within a specified time limit.

The best air strike plan minimizes the probability of the aircraft being detected by the radars and, at the same time, maximizes the fuel remaining upon arrival at the target so that the enemy fighter aircraft can be engaged or evaded.

The purpose of this material is to acquaint you with the:

1. Detection ability of multiple enemy radars when there is overlapping detection coverage between radars in proximity to each other
2. Means of measuring the goodness of an air strike plan
3. Characteristics of the nonlinear programming optimization technique.

The training goals are to:

1. Develop expertise in using the equipment
2. Develop a feel for the best way to work with the computer to find the best air strike paths and speeds.

In training you will do eight problems with the optimization technique. Experimental data collection will then be done for twelve problems. Thus, you will do a total of 20 problems. Each problem will last 15 minutes.

A. REPRESENTATION OF ENEMY RADAR DETECTION CAPABILITY

The capability of a single enemy radar to detect your aircraft is represented by concentric circles around the radar location. Detection capability is the same at all points on each circle and is a specified percentage of the peak detection capability of the radar. (See Figure C2.) Notice that as you go along a radial line toward the center of the concentric circles, detection capability increases up to 90% of the peak level. The peak occurs between the two 90% circles, and detection capability decreases from the peak as you get closer to the radar location. Thus detection capability may be visualized in three dimensions as a volcano with a rim and a crater in the center of the volcano. The "Detection volcano" is centered on the radar's position.

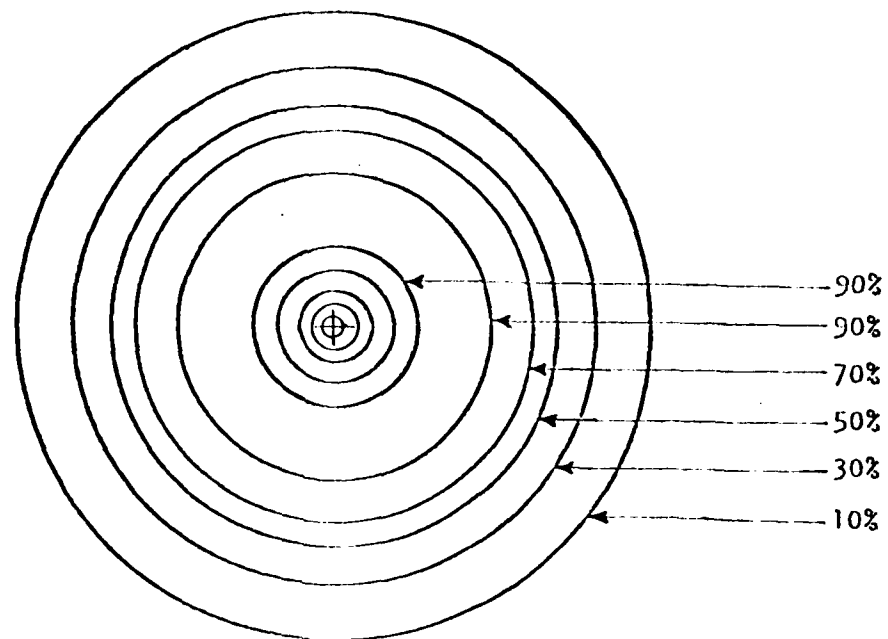


Figure C2 . Single Sensor Coverage Template. (Circles show distance from sensor location, center, at which percentage values of the peak detection rate occur.)

When several radars have overlapping coverage as shown in Figure C3, the probability of detecting your aircraft at a point within areas of overlap is higher than it would be at that same point if only one radar could detect at that point. Thus there is a joint detection capability throughout areas of overlap. The points where joint probabilities of detection are equal are connected together to form contours as shown in Figure C4. The contours have the same general meaning as the concentric circles in Figure C2, that is, each contour is the set of points where detection capability is some specified percentage of the peak joint detection capability. The set of contours is analogous to a topographical map. The difference is that each contour on a topographical map corresponds to an altitude above sea level and each detection capability contour corresponds to a detection capability between zero capability and the peak capability.

B. MEASURING THE GOODNESS OF AN AIR STRIKE PATH: THE UTILITY FUNCTION

The problem is to select an optimal air strike path, so an appropriate utility criterion function is one which measures the "goodness" of an air strike path. The two variables selected to determine the goodness of an air strike path are fuel consumption along the path and probability of being detected by one or more enemy radars prior to reaching the target. The utility criterion function incorporates a tradeoff between minimizing the probability of being detected by enemy radars on one hand and maximizing the fuel remaining upon arrival at the target on the other.

The utility function takes on any value between 0 and 100, with higher utility values corresponding to "better" paths. A family of parameterized curves from the utility function is shown in Figure C5. The figure shows that as the probability of being detected by enemy sensors decreases, the utility value goes up. Also, if the probability remains constant, the utility value increases as fuel consumption drops. It is obvious why it is desirable to minimize the probability of being detected by enemy sensors. The rationale for encouraging fuel preservation is that if detection occurs at any time up to arrival at the target, there should be as much fuel left as possible in order to do some flight maneuvering to try to return safely.

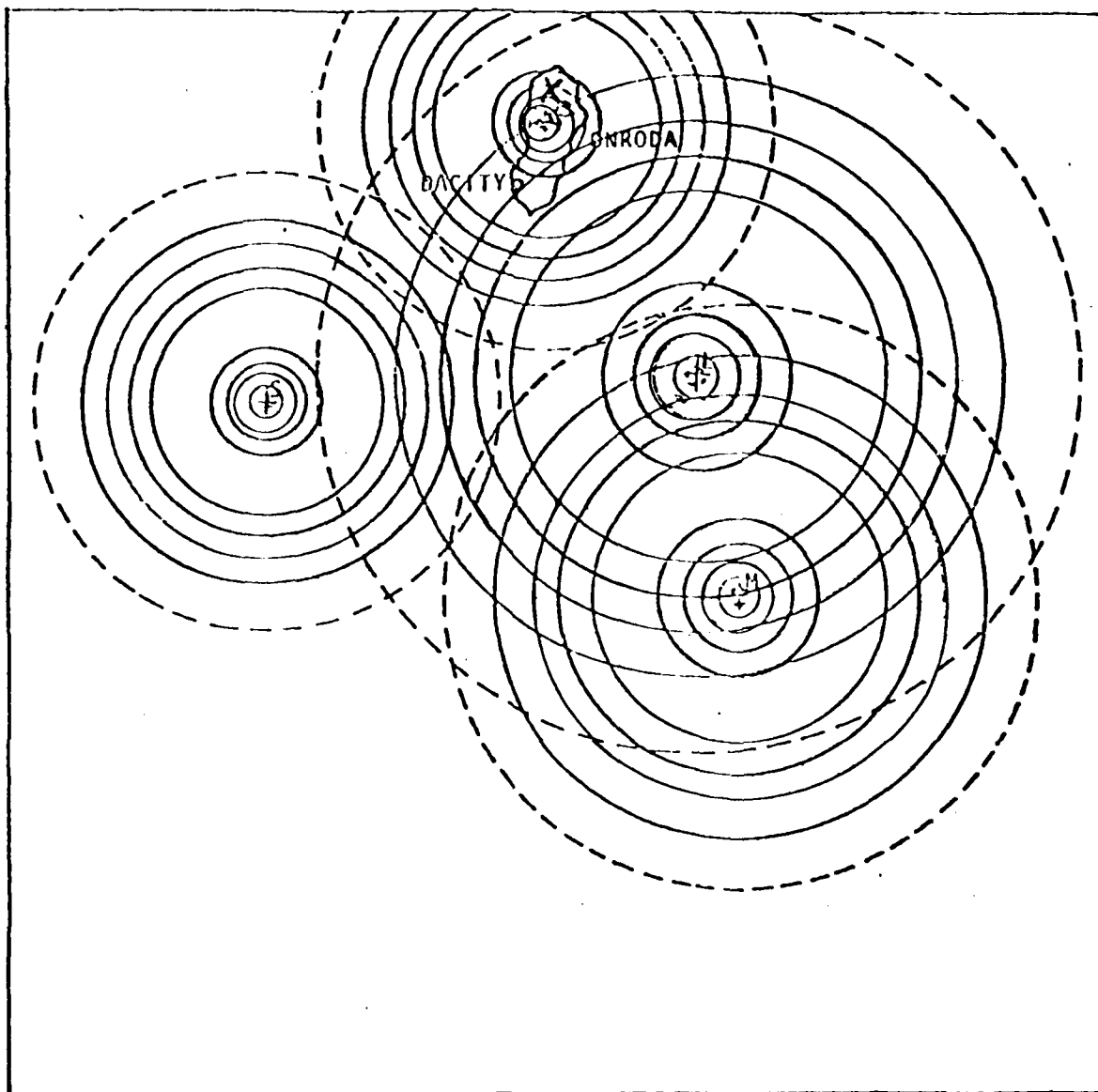


Figure C3. Display with Four Sensor Coverage Templates Shown.

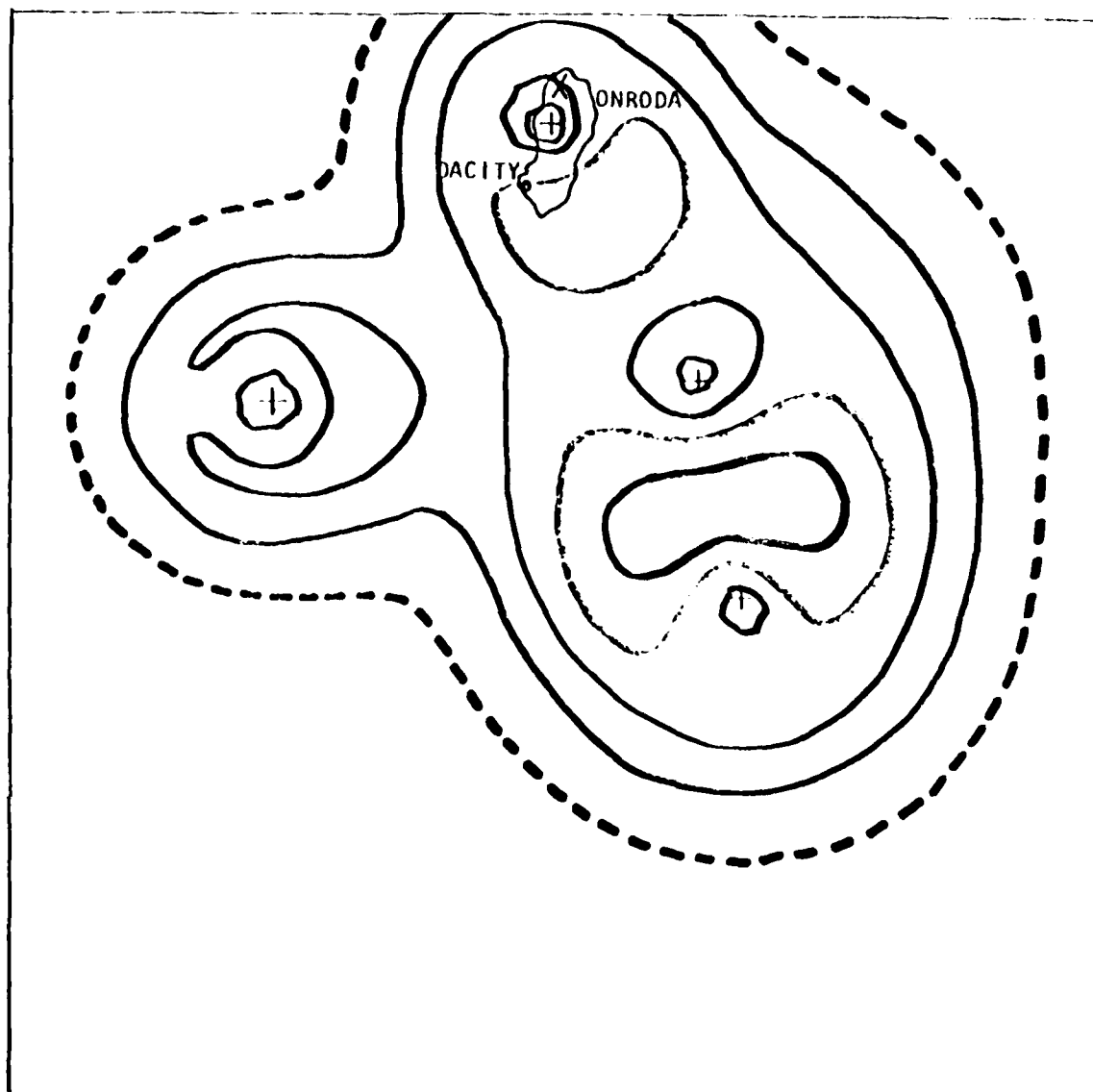


Figure C4. Contours Showing Joint Probabilities of Detection for Four Radars Located at Positions Marked with a Cross (+).

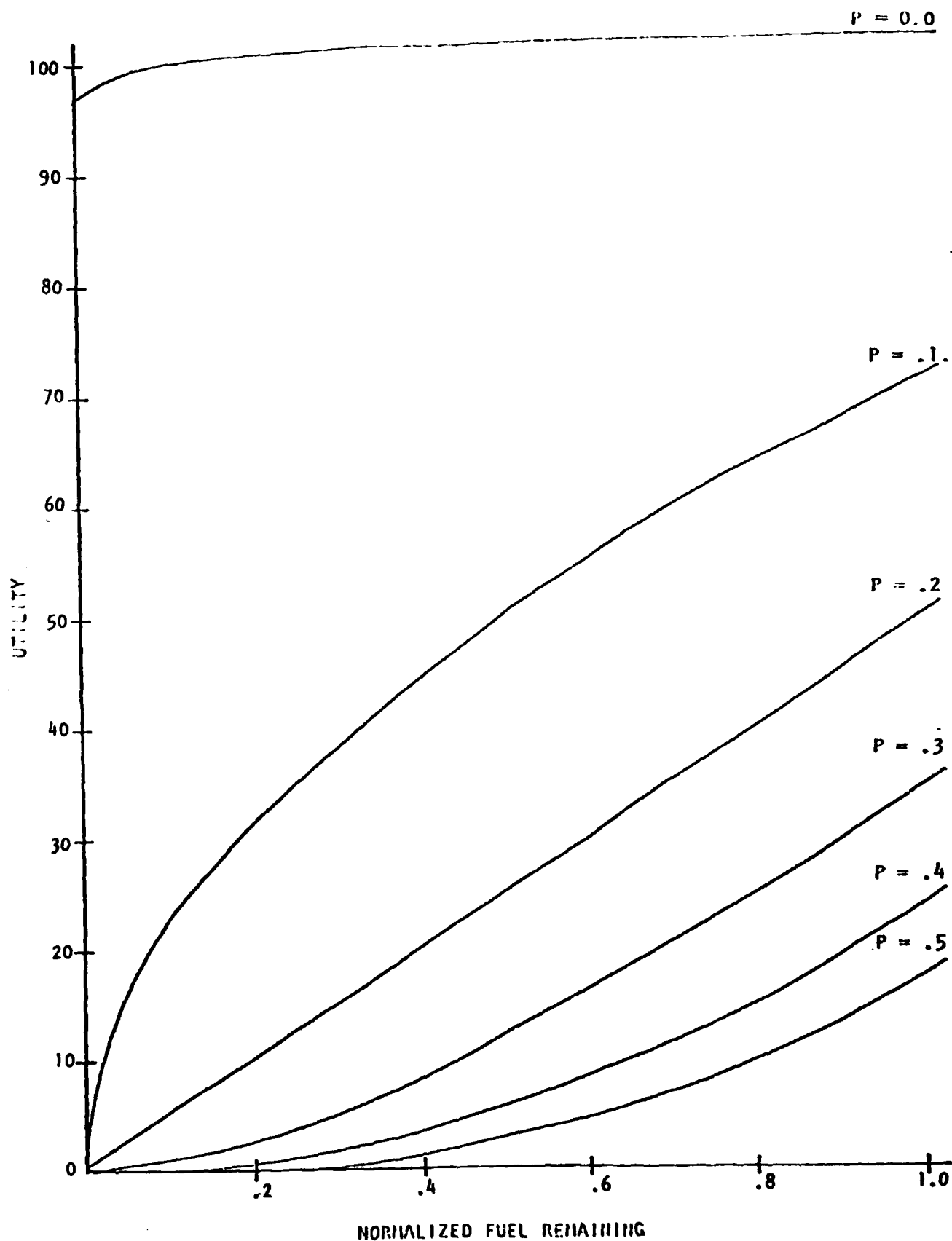


Figure C5. Family of Parameterized Utility Function Curves.

In general the two goals of minimizing fuel consumption and minimizing the probability of being detected are incompatible. A nontrivial optimal air strike path thus requires a reasonable compromise between the two goals.

C. CHARACTERISTICS OF THE NONLINEAR PROGRAMMING OPTIMIZATION TECHNIQUE USED IN THE EXPERIMENT

The setup for the nonlinear programming (NP) technique includes the starting "point" or first trial solution. In the air strike problem, the starting "point" is (a) five path legs connecting the air strike launch point and the target and (b) speeds for each leg. The legs are specified by picking four "way points" between the launch point and target. Speeds are selected from a range of 250 to 1000 knots. After the start point has been specified, the NP technique operates to find a better combination of way points and speeds. It does this by exploring changes in the location of each way point and the speed for each leg. Improvement in the air strike path takes place slowly over many explorations, i.e., trials. An advantage of NP is that it considers all the points in a geographical region instead of just a set of grid points and all speeds instead of just a few. A disadvantage is that the "solution" will be best for the region explored but that better solutions may exist in unexplored regions and the NP technique is unable to direct itself to look in these unexplored regions. In optimization jargon, NP may find a local optimum but not the global optimum.

D. OPERATION OF THE NONLINEAR PROGRAMMING OPTIMIZATION

At the beginning of a problem the display will appear as shown in Figure C6. The path from launch point to target is a straight line with way points indicated at $1/5$, $2/5$, $3/5$, and $4/5$ of the straight line distance. Speed for each leg is initially set by the program at 600 knots as indicated on the plot at the left in Figure C6. The subject uses the appropriate buttons on the function button box (see Figure C7) and the joystick to change the position of the four way points. He uses the appropriate function buttons and the keyboard to change speed on any leg.

The subject's purpose is to direct the NP technique to investigate as many reasonable potential solution regions as possible in 15 minutes.

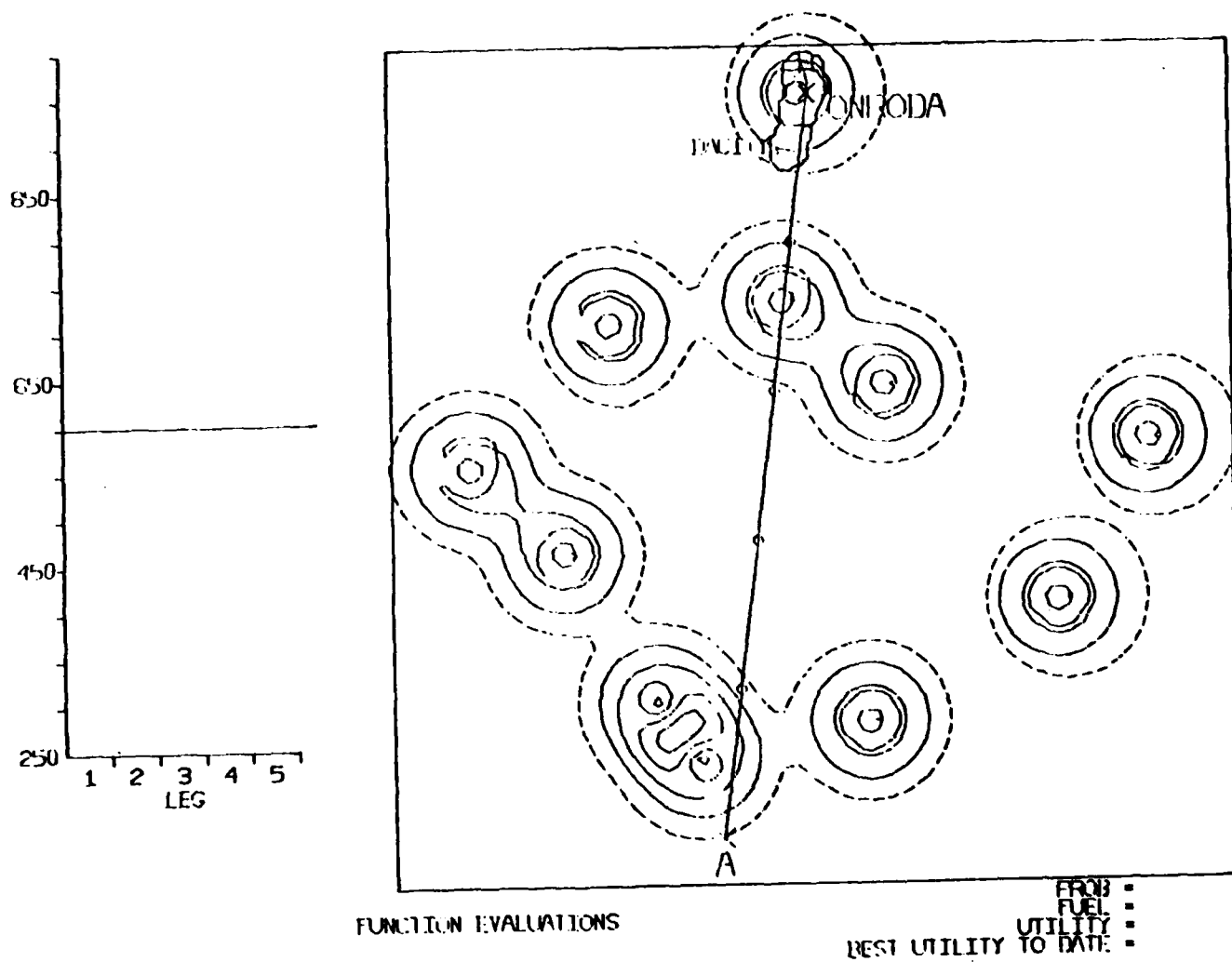


Figure C6. Display Appearance at Beginning of Problem to be Solved Using NP Algorithm.

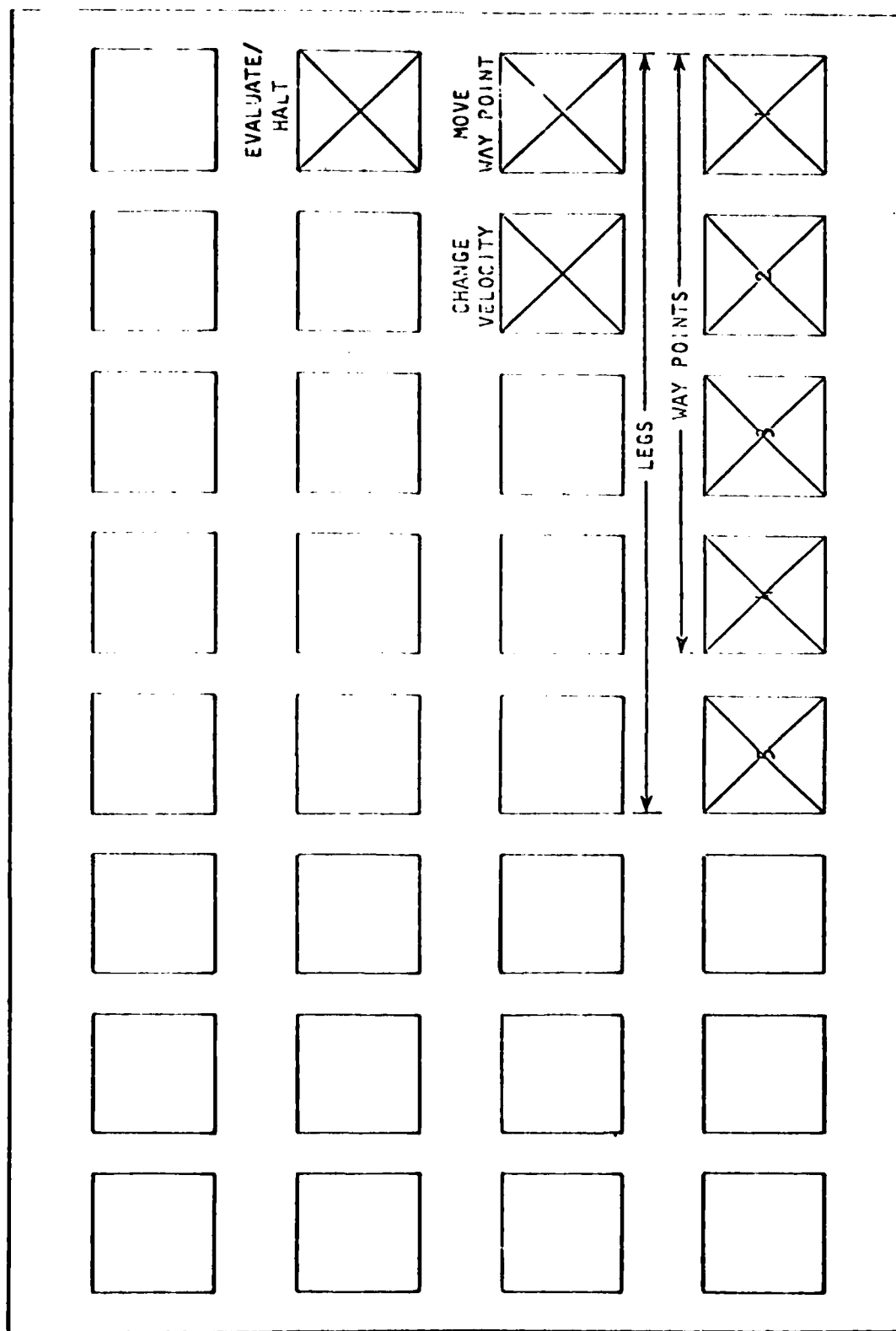


Figure C7. Function Button Configuration for the Nonlinear Programming Optimization.

As soon as the problem is shown on the display, the subject must decide what region he wants to explore first. He is to pick the region that he thinks is most likely to contain the best solution. He then changes the locations of the way points and speeds prior to starting the NP algorithm. At the beginning of the problem the three buttons designated as "Evaluate/Halt," "Change Velocity," and "Move Way Point" are lit on the box. In order to move a way point, push that button. When this is done the four buttons marked 1, 2, 3, and 4 will light. Then push the button corresponding to the point to be moved, i.e., 1, 2, 3, or 4. Way point 1 is the closest to the beginning of the strike path and 4 is the nearest to the end (ONRODA Island). Moving the way point is accomplished with the joystick. When a single way point is changed, a second way point can be changed by pressing "Move Way Point" and the appropriate number of the way point. The act of pressing "Move Way Point" records the position of the last way point that was changed.

To change a speed on one of the five legs, push "Change Velocity." The five buttons marked 1, 2, 3, 4, and 5 will light. Leg 1 refers to the leg closest to the path start point and leg 5 refers to the path closest to the end point (ONRODA Island). Then push the button corresponding to the leg for which you want to change speed and:

1. Use the teletype keyboard to input the speed you want used on the selected leg. Put a decimal point at the end of the number. (This is essential.)
2. Push the teletype key marked "CR."

Thus, if you wanted to change the speed on leg 3 to 850 knots, you would:

1. Press function button "Change Velocity"
2. Press function button "3"
3. Press teletype key "8"
4. Press teletype key "5"
5. Press teletype key "0"
6. Press teletype key "."
7. Press teletype key "CR"

When you have changed all the way points and speeds to those you want, then press the function button marked "Evaluate/Halt." The NP algorithm will begin to operate, i.e., "Evaluate," using your starting point consisting

of the four way points and five speeds. Once the algorithm has begun operating, only the "Evaluate/Halt" button will remain lit and the only control at the operator's disposal is to halt operation by pushing this button.

The primary indicators that the operator uses to decide whether to halt the algorithm are the displays of the number of function evaluations and the utility of the latest trial solution. In general, a plot of utility versus function evaluations would appear as shown in Figure C8. The subject should stop the algorithm when it reaches the point shown in Figure C8 because there will be little more utility to be gained by letting the algorithm continue. He should then input a new set of way points and speeds and start the algorithm again.

As the algorithm operates you will note variable length arrows appearing briefly at each way point. These represent potential changes in the location of a way point being considered by the algorithm. When utility levels off, the magnitudes of changes in the following will also become small:

1. Value of "Prob," i.e., the probability that the air strike will be detected prior to arrival at the target.
2. Value of "Fuel," i.e., the fuel that will be consumed for the latest trial solution.
3. Speed changes indicated on the speed/leg graph.
4. Lengths of arrows appearing at each way point.

While the algorithm is operating on the first set of way points and speeds input by the operator, he should count the number of regions that could reasonably be expected to contain the best path. Dividing 15 minutes by the number of regions to be explored indicates approximately the number of minutes the operator should devote to each region. Depending on the problem, there will be enough time to explore 3, 4, or 5 regions.

At the end of 15 minutes the computer will have stored:

1. The utility of the path comprised of the first way points and leg speeds entered by the operator.
2. The utility of each best-solution-to-date at the end of each minute, excluding the first minute.

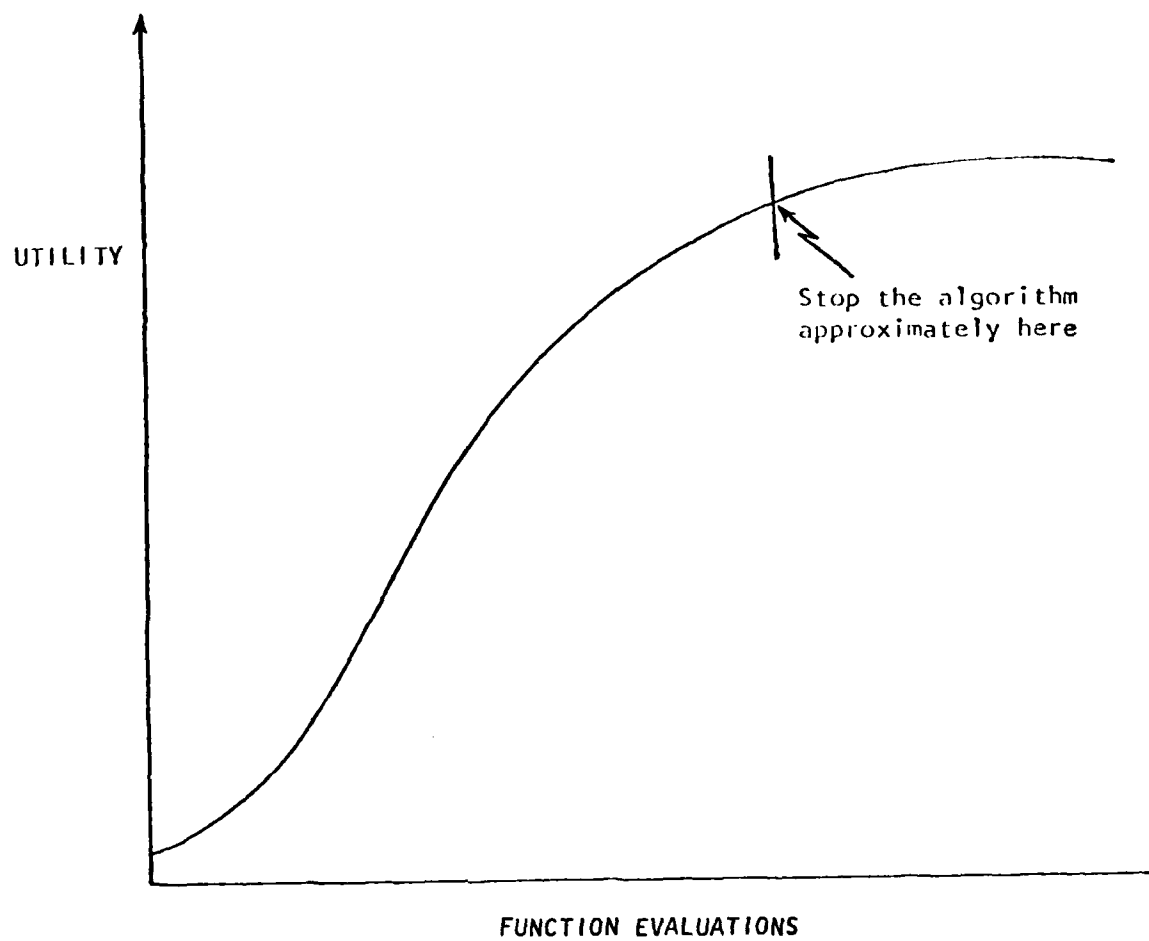


Figure C8. Typical Plot of Utility Versus Function Evaluations.

These are the data that will be used in the analysis of operator generated data.

E. GUIDELINES

There are two types of data being analyzed:

1. Utility of the path comprised of the first way points and leg speeds entered by the operator. Thus the operator's first goal is to do the best he or she can on this.

2. Operator performance will be calculated at the end of each trial by adding the 14 utilities of the best-solution-to-date at the end of each minute (excluding the first minute) and dividing this sum by 14. Thus, operator performance for the entire trial is the average of the 14 utilities. The operator's second goal is to maximize this average. In general this is done by exploring the regions which could contain the best path in the order of the estimated likelihood that each contains the best path. This is compatible with the operator's first goal because, if the operator is correct concerning the region which contains the best path, then the average utility will be nearly equal to the utility of the best path. This is true because the computer only stores the best utility to date and will therefore not store the utilities of paths investigated after the first if the first region explored contains the best path.

Other general rules to be used with the NP technique are:

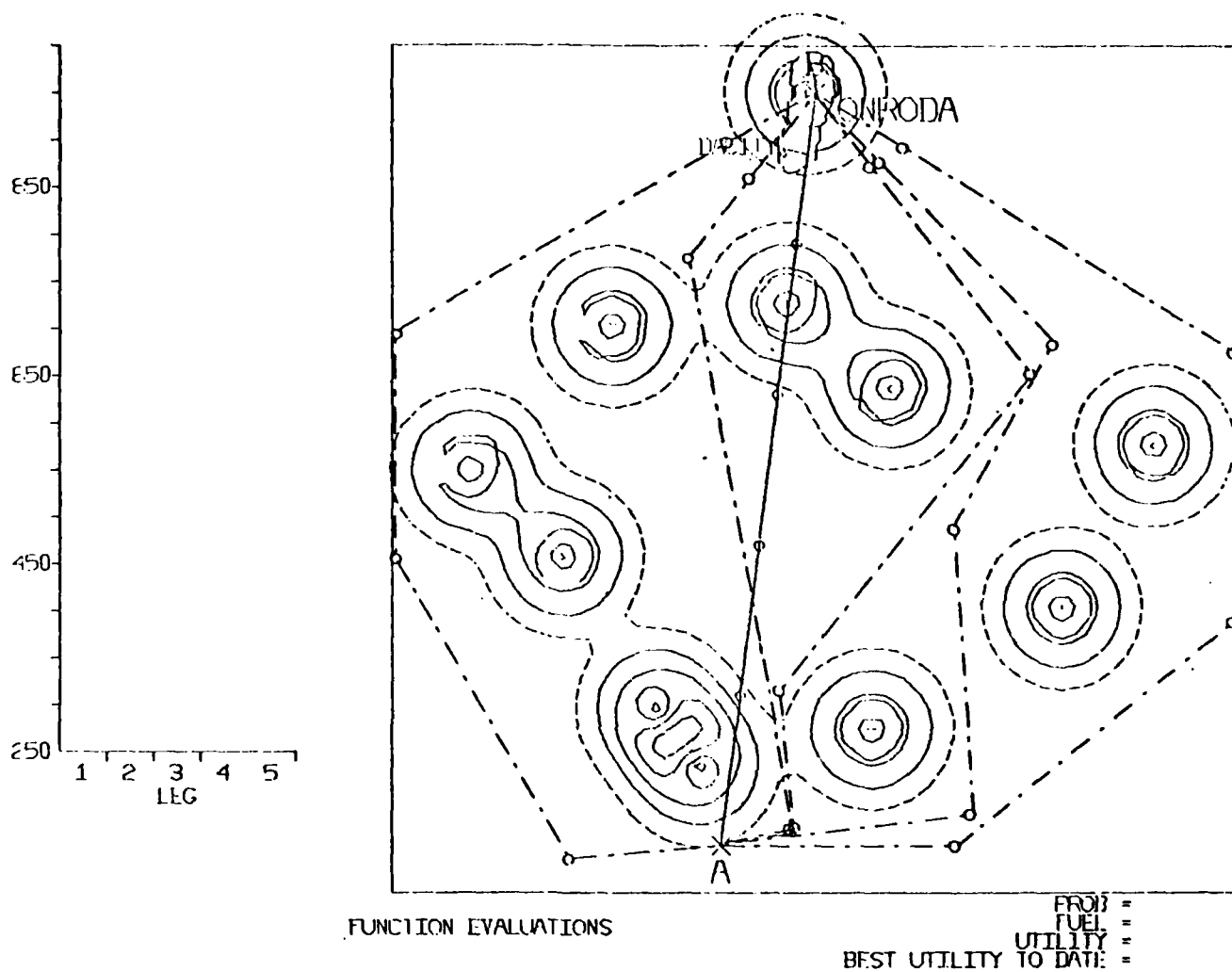
1. Those portions of a path that are completely outside the detection contour should be transited at low speeds.

2. Those portions of a path that traverse a high detection probability contour should be transited at high speeds. In particular, the last leg of the path to the target should be transited at high speed since it must pass through the high detection region around ONRODA airport. It is best to locate the last way point just outside this region and use a speed such as 999.

3. Paths should be drawn to pass through low detection probability regions. However, a completely roundabout path that avoids detection contours completely is not a sure winner because long paths use a lot of fuel.

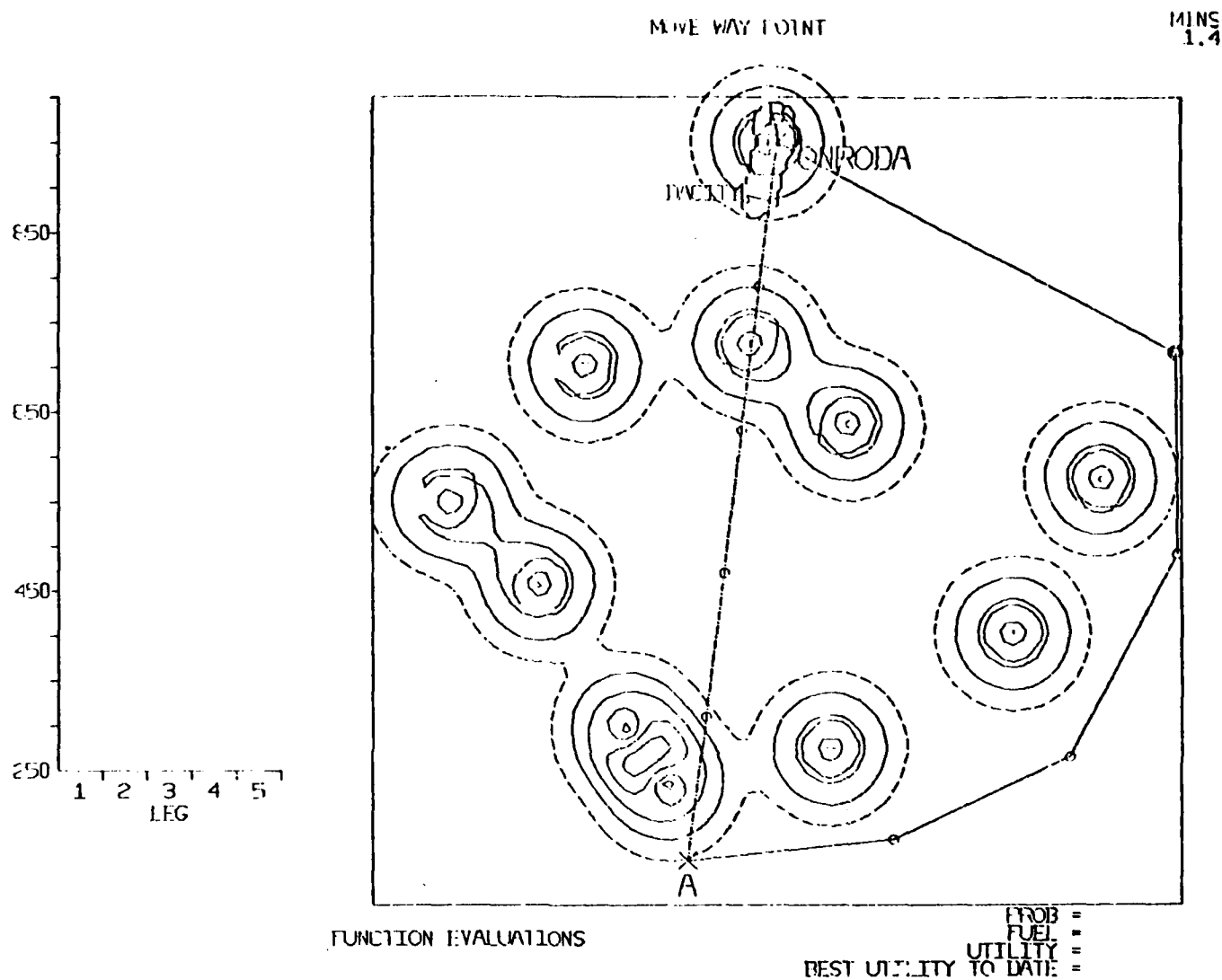
4. When crossing detection regions, it is a good idea to place way points on both sides of the region, just outside the lowest detection probability contour.

The following nineteen plates illustrate these points using a sample scenario. Note that the speed/leg graph shown on the left of each plate is from an earlier version of the program; it has been replaced by a simple table of speeds.



This is how the display appears at the beginning of the problem. Five potential best paths are shown as dot-dash-dot lines.

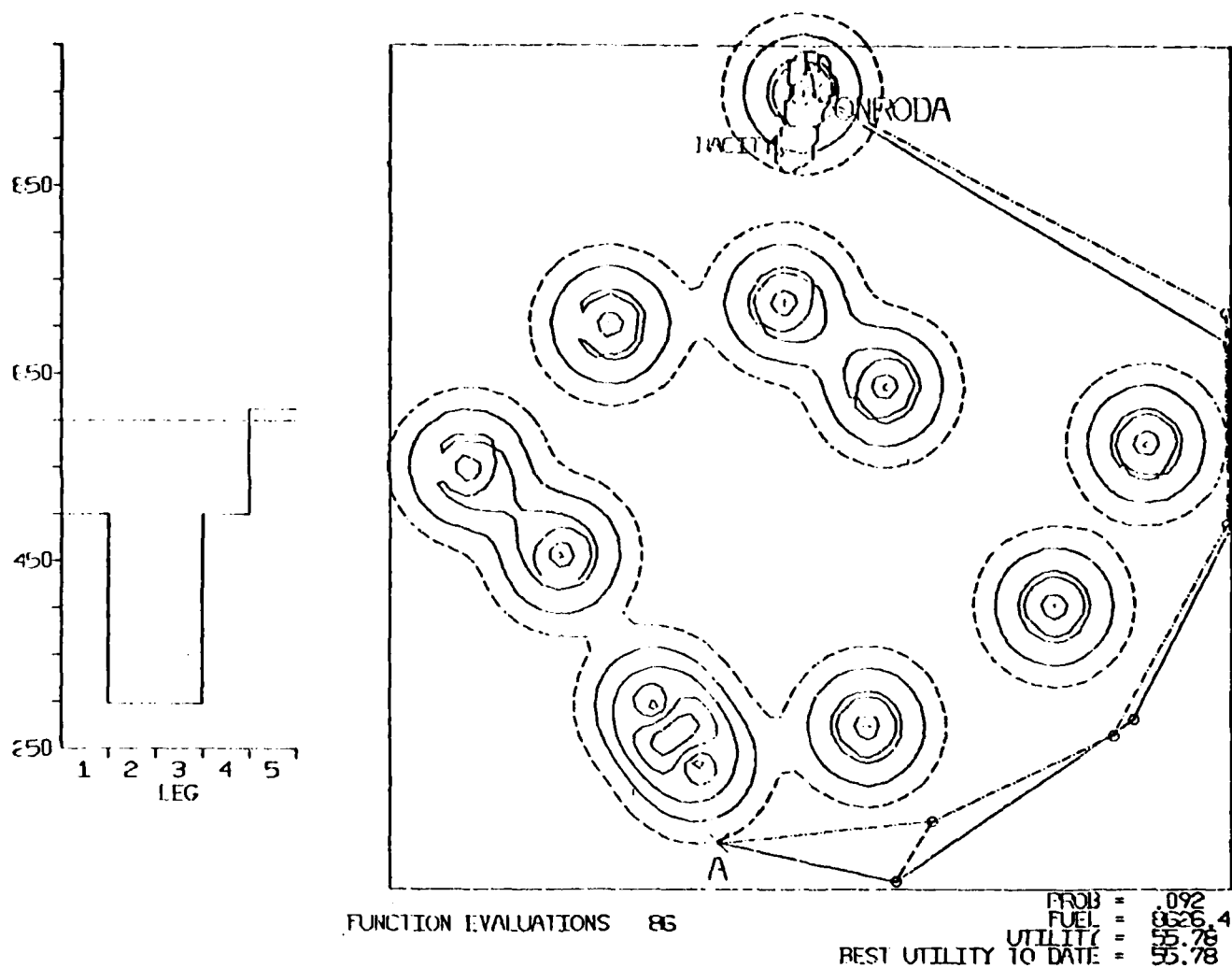
Figure C9. First Plate, Example Problem.



The operator chose to explore paths from right to left. It would have been better to have configured the path so that the last leg began just outside the contours around ONRODA. The previous starting path remains on the display as a dot-dash-dot line.

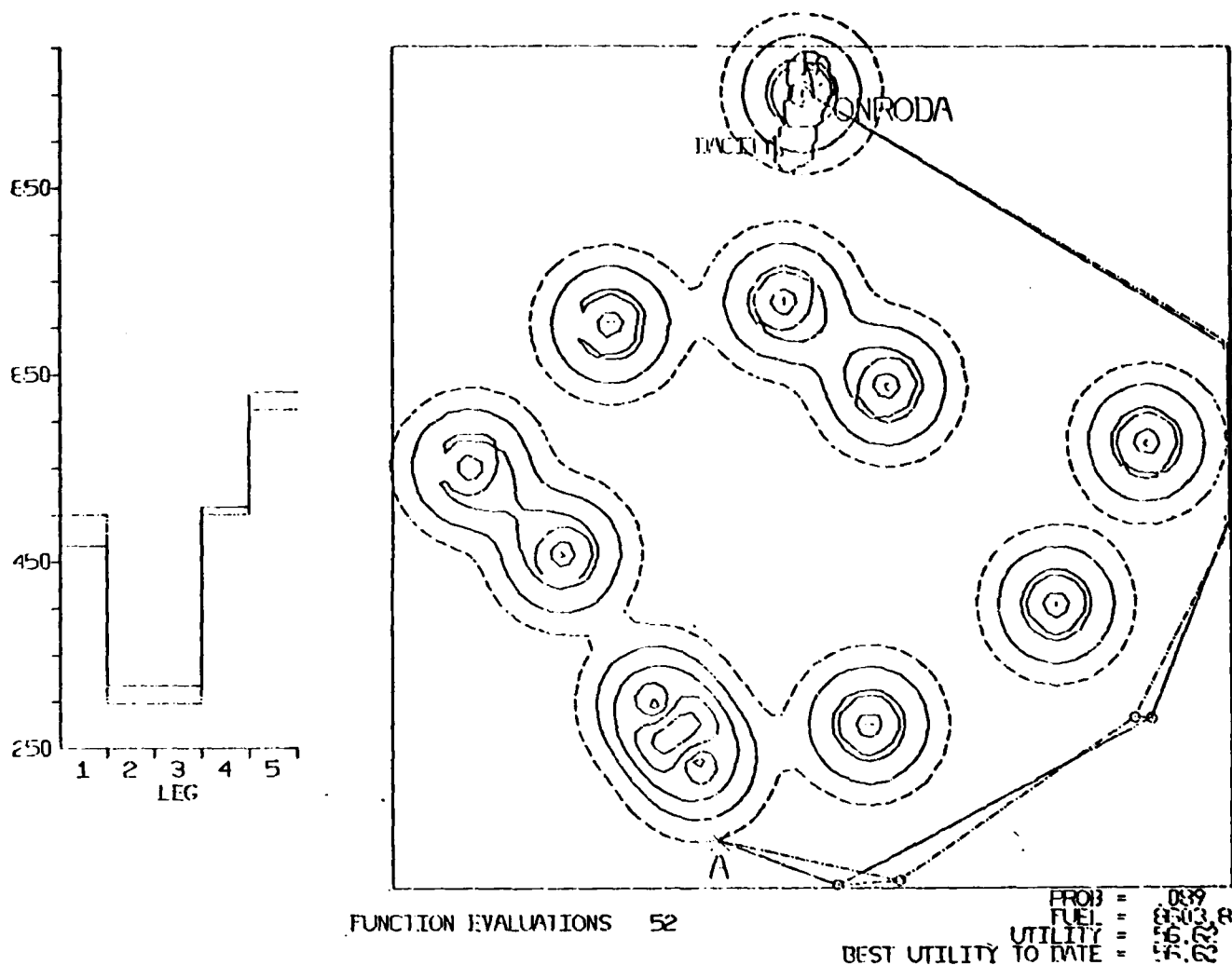
Figure C10. Second Plate, Example Problem.

MINs
2.6



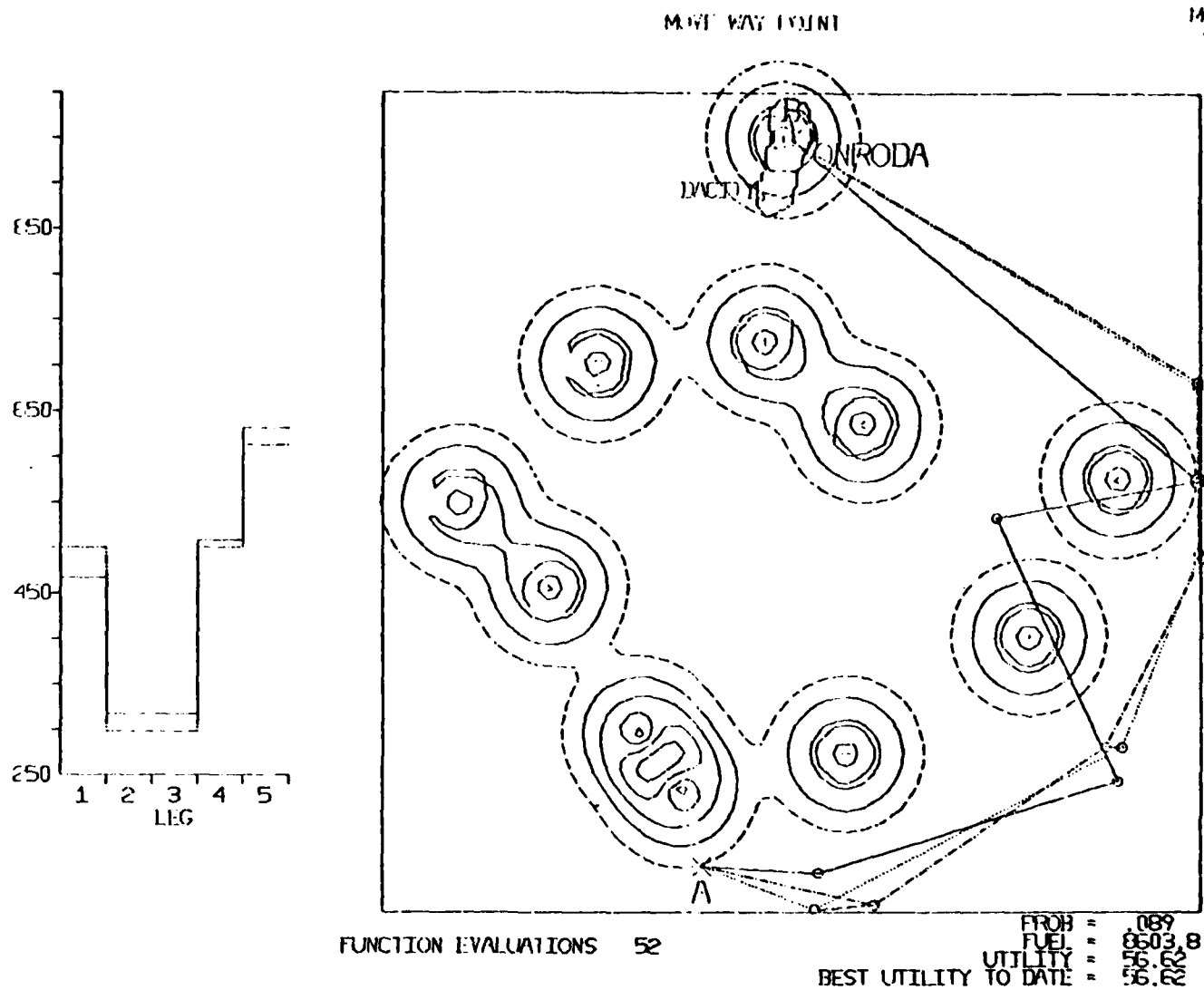
The operator stopped the algorithm at the end of 86 evaluations in order to get this picture. Note that the solution moved the first way point down in order to get away from the contours above the point.

Figure C11. Third Plate, Example Problem.



The operator restarts the algorithm without making any changes. At the end of another 52 evaluations (138 total), the operator stops the algorithm because (a) the step sizes being considered are very small and therefore the possible utility improvements will also be small, and (b) the utility hasn't increased very much in the last 25 or so evaluations.

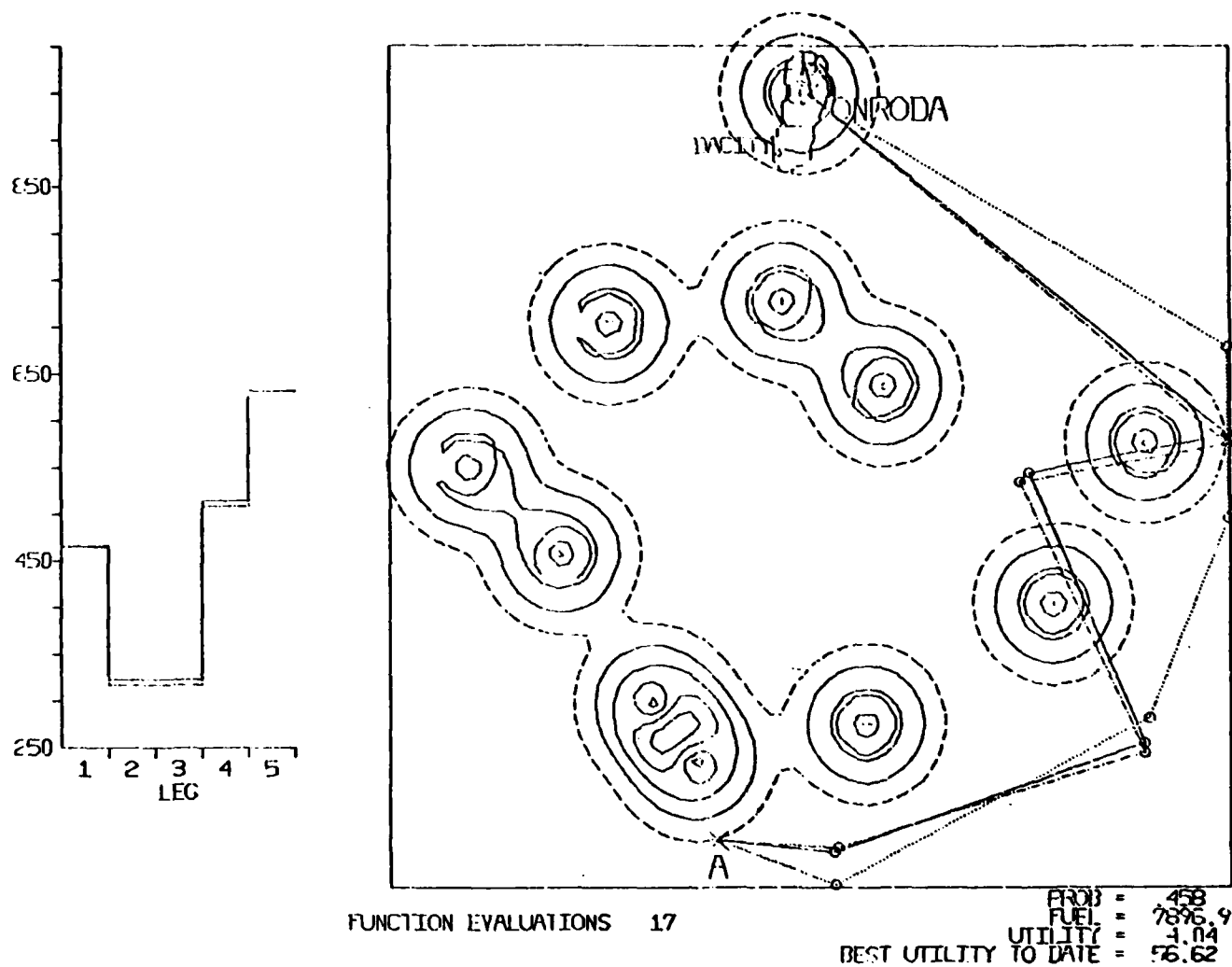
Figure C12. Fourth Plate, Example Problem.



The operator puts the third and fourth way points in an illogical combination of places and makes small adjustments to the other two way points. The point will be to see what the algorithm does.

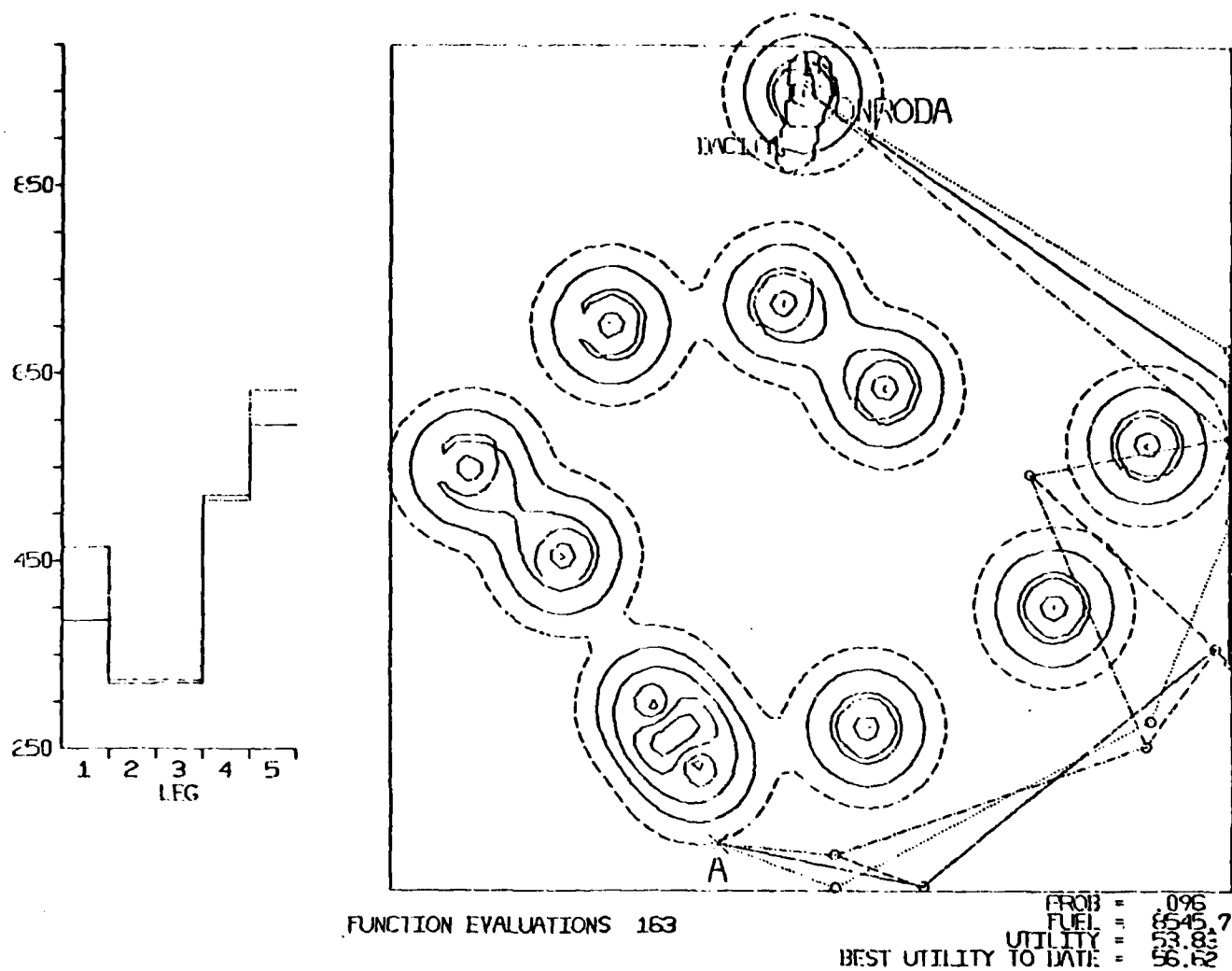
Figure C13. Fifth Plate, Example Problem.

MINS
5.4



At the end of only 17 evaluations not much has happened.

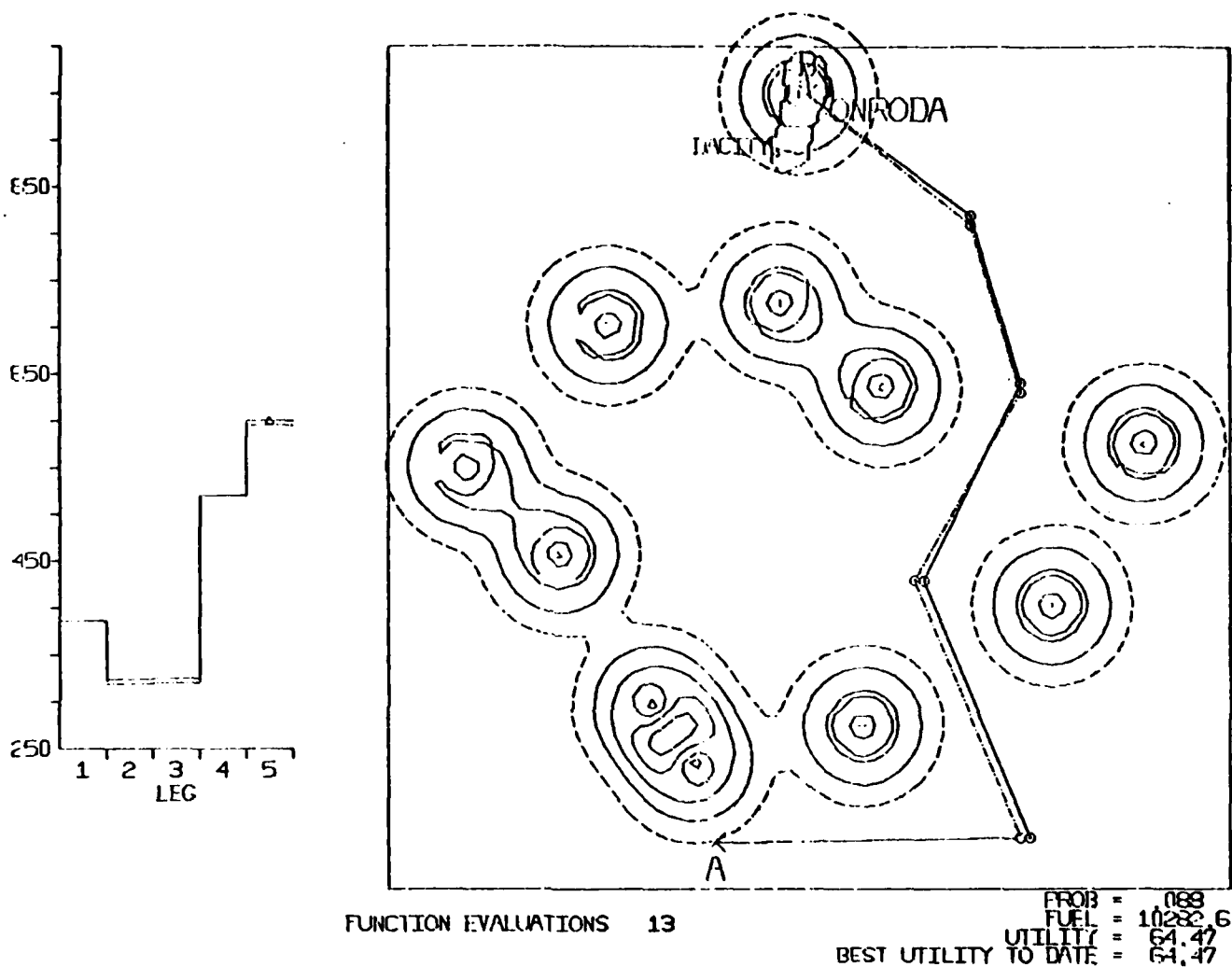
Figure C14. Sixth Plate, Example Problem.



At the end of 163 evaluations the algorithm has found its way over to a much better position for the third way point but the utility is not as good at 163 evaluations (53.83) as it was at 140 evaluations with the earlier, better selection of way points (56.62 for the starting path of Figure C10).

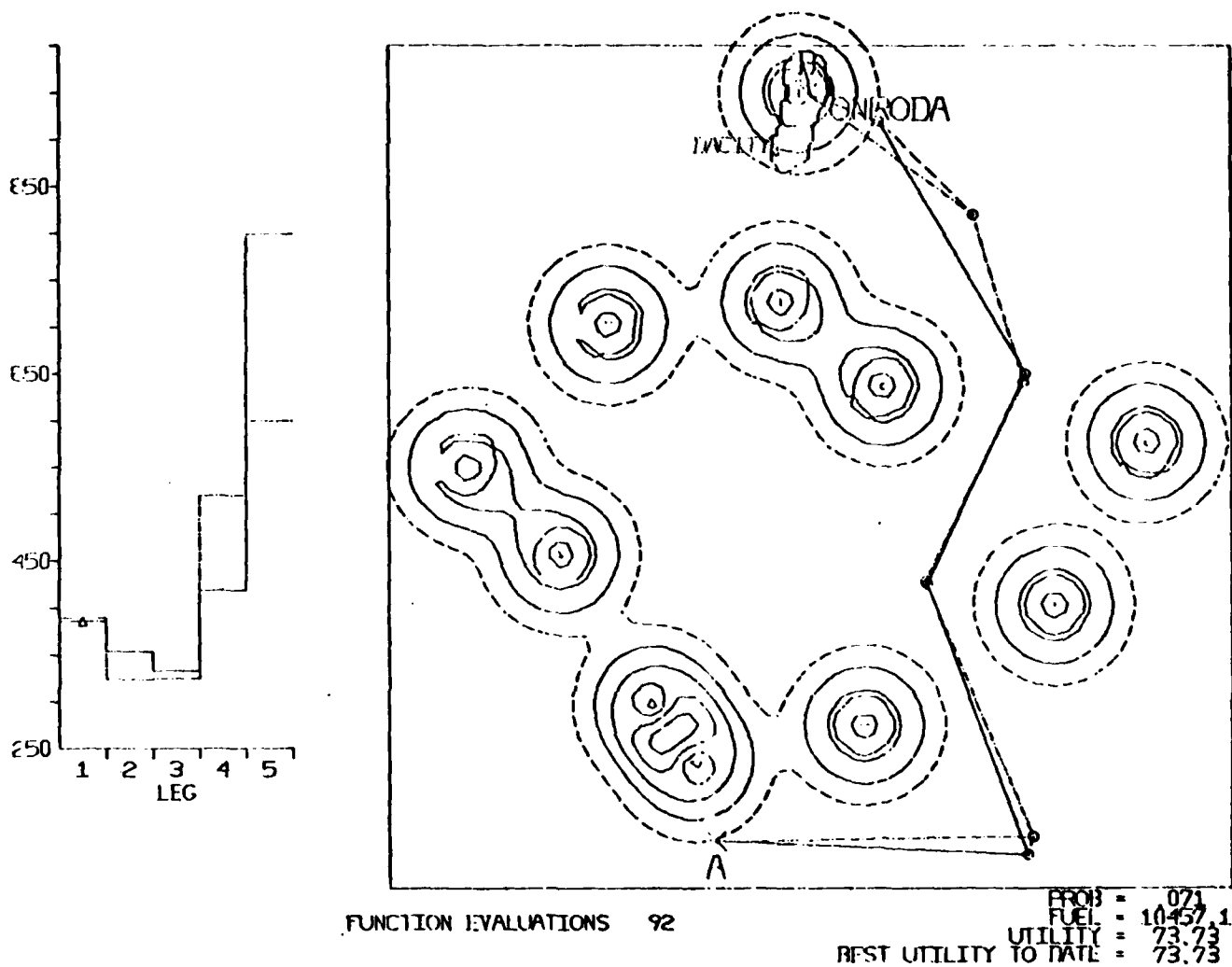
Figure C15. Seventh Plate, Example Problem.

MIN
8.6



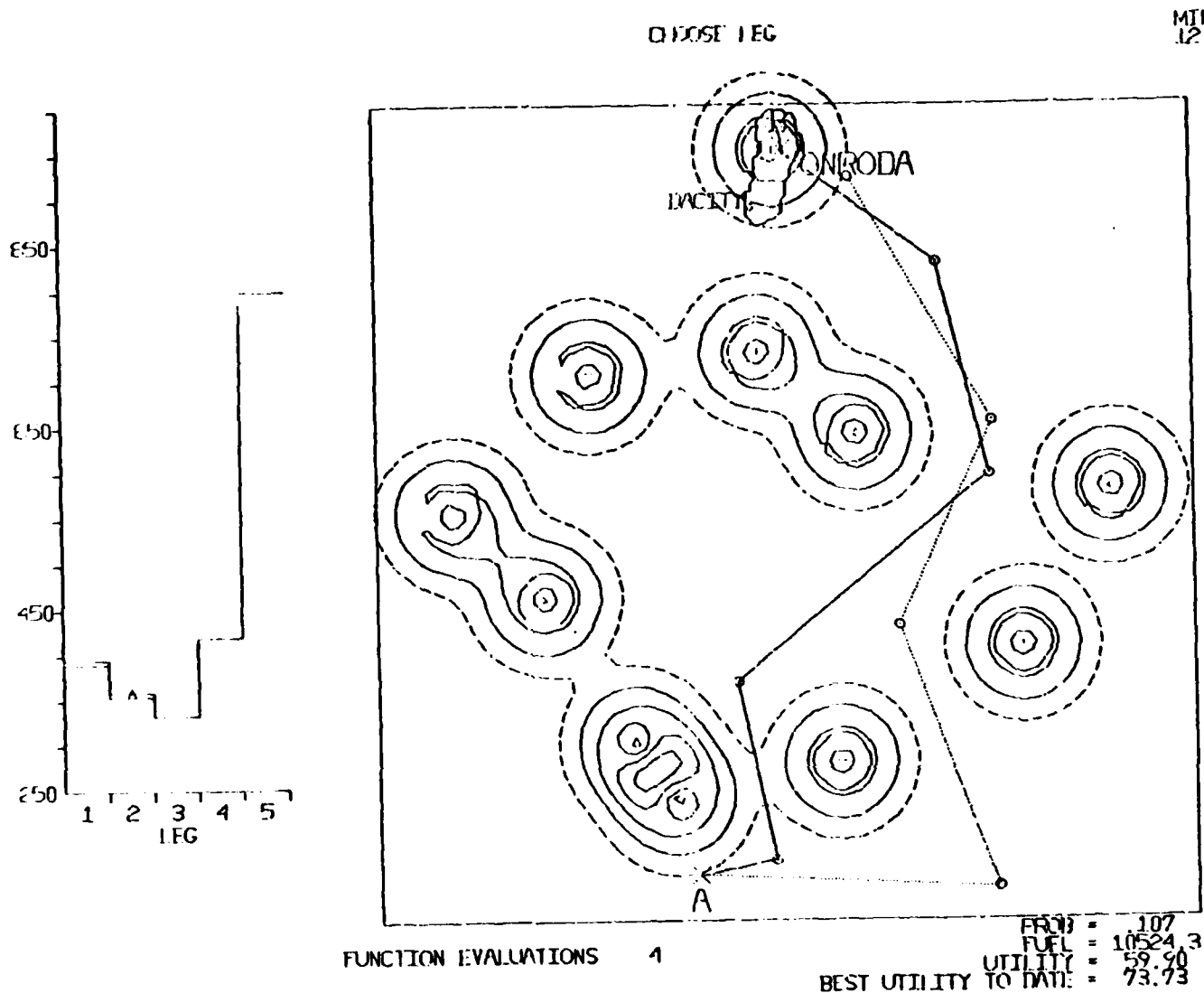
The operator selects a new set of way points and the algorithm begins to explore around these. Again, he should have placed the last way point closer to ONRODA.

Figure C16. Eighth Plate, Example Problem.



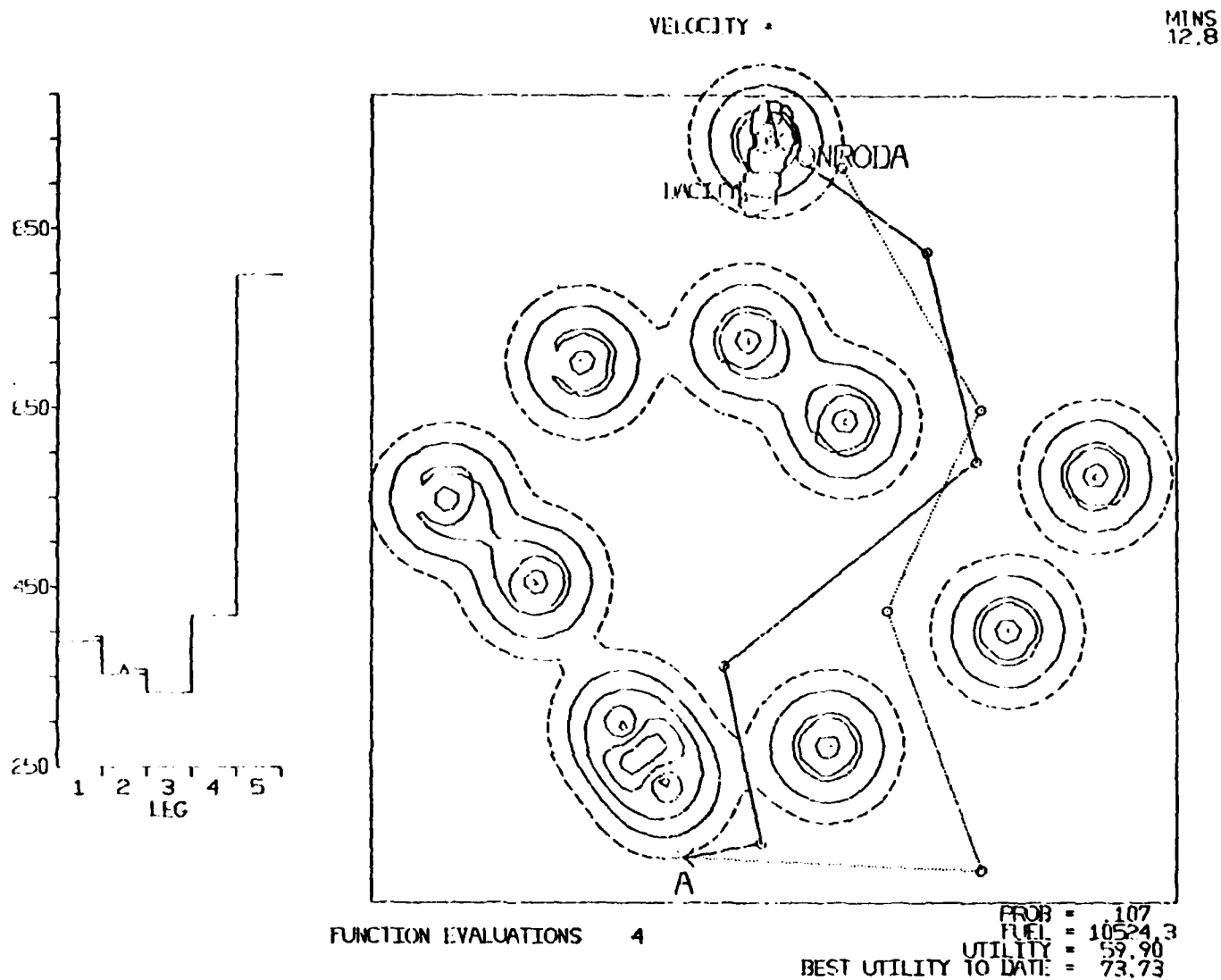
At the end of 92 evaluations the operator stops the algorithm. Note that the algorithm has moved the last way point much closer to ONRODA and has greatly increased the speed for the last leg.

Figure C17. Ninth Plate, Example Problem.



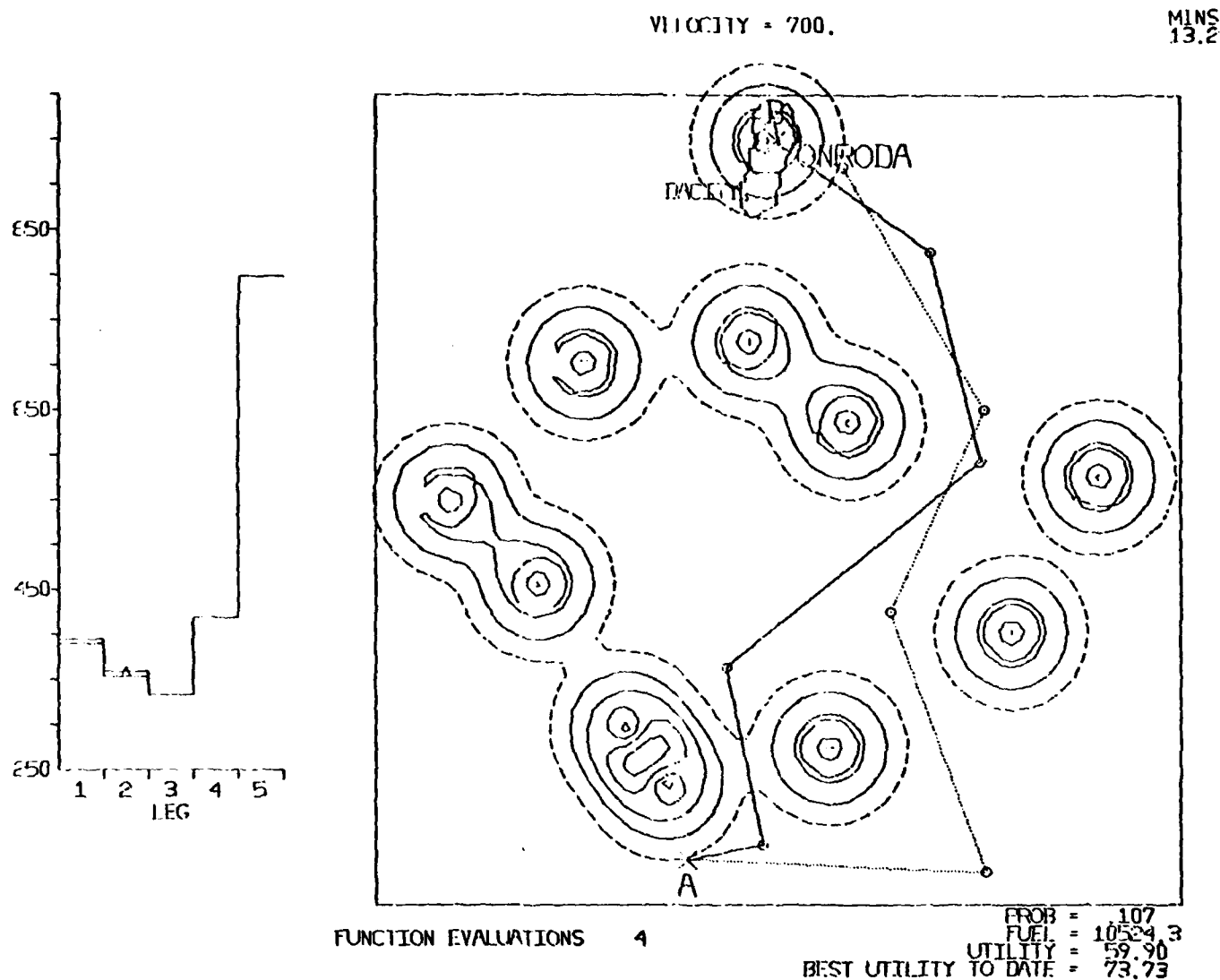
The operator has already selected way points for the third path to be explored by the algorithm. Utility is 59.90 at the end of four evaluations, and then the operator stops the algorithm. He has decided to change the speed on a particular leg and accordingly pushed the "Change Speed" function button. The prompt "Choose leg" then appears at the top of the display. Then he pushes the function button corresponding to the desired leg.

Figure C18. Tenth Plate, Example Problem.



Immediately the prompt "Velocity = " appears at the top of the display.

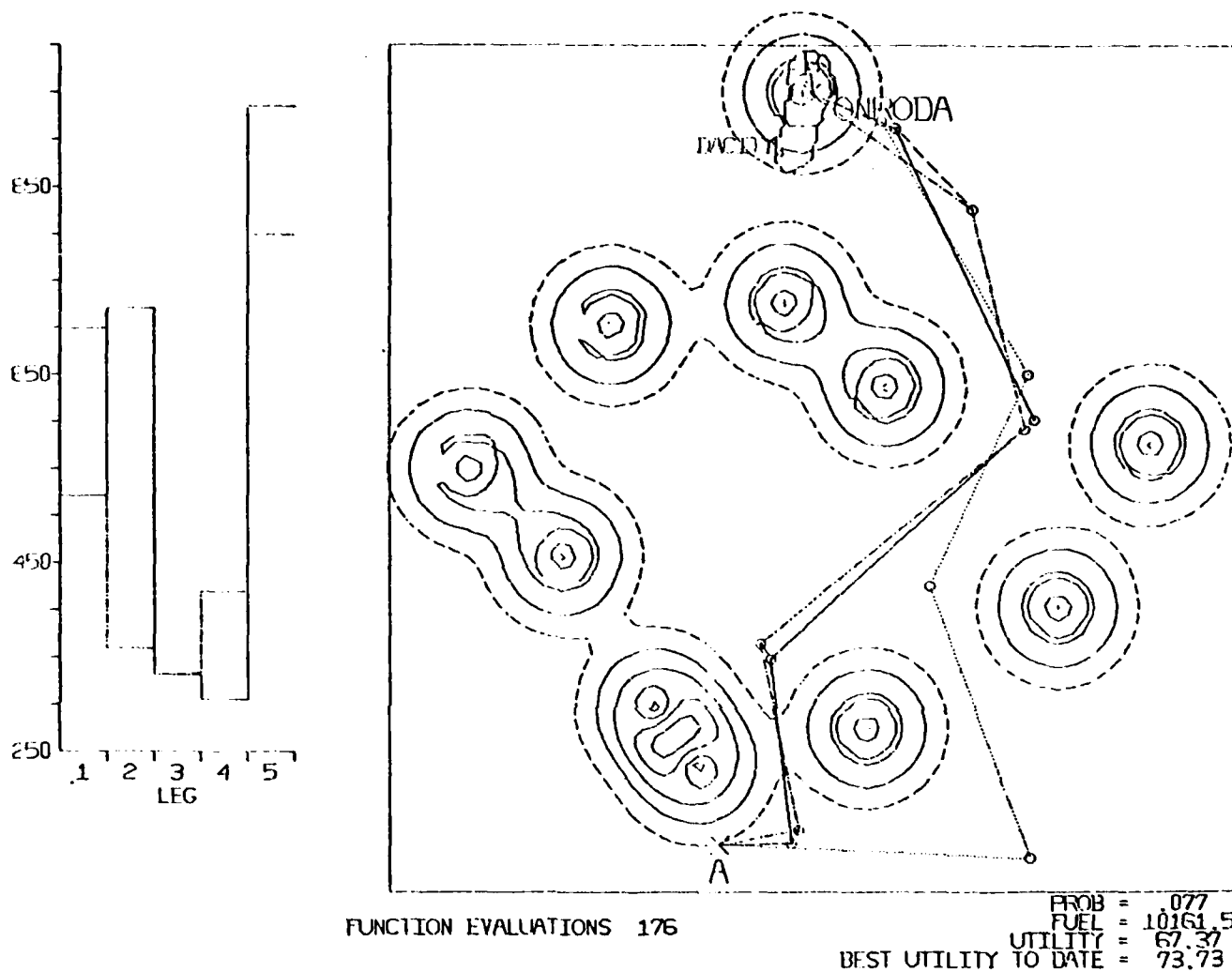
Figure C19. Eleventh Plate, Example Problem.



The operator then types "700. CR " and 700 appears at the top of the display. The operator restarts the algorithm.

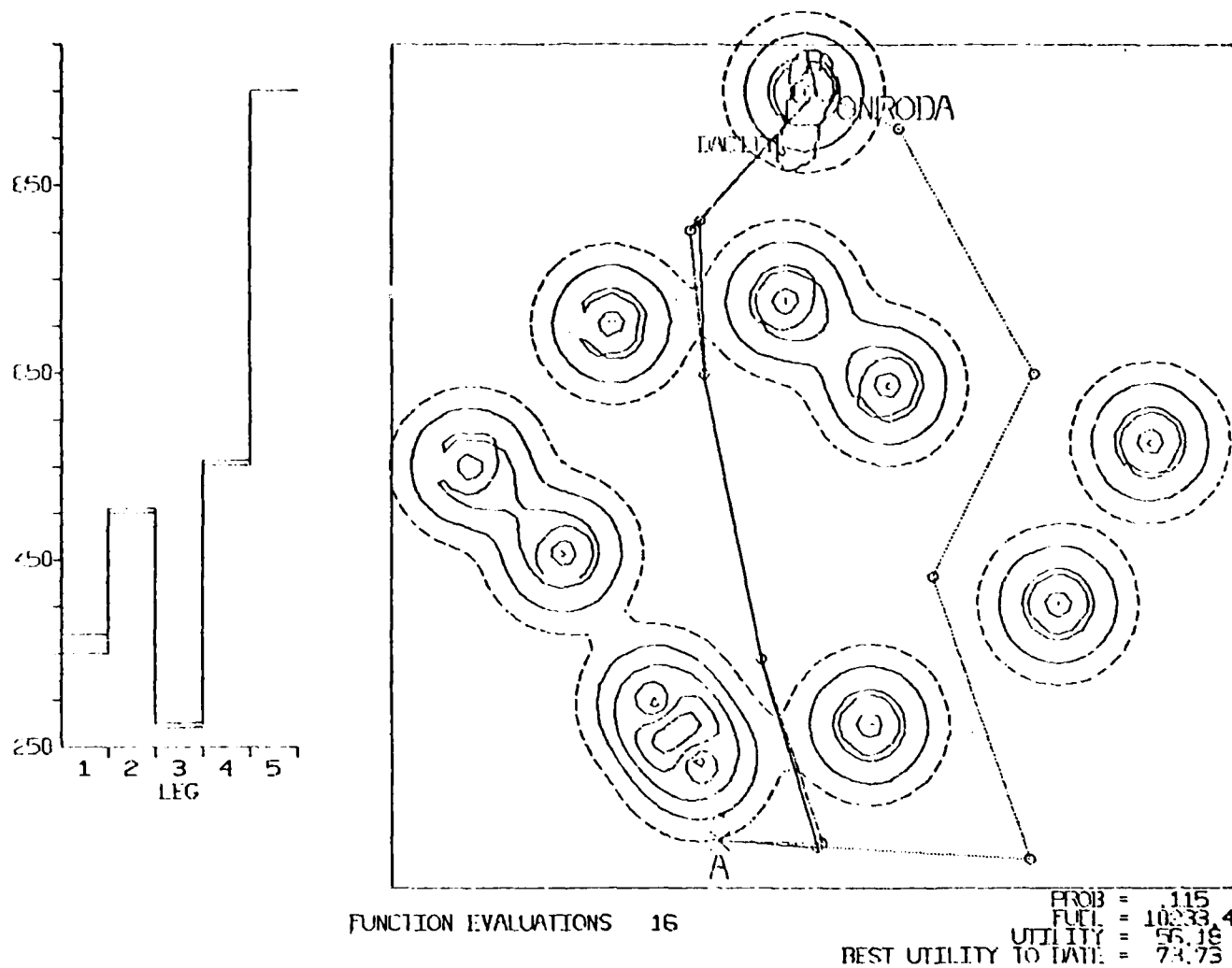
Figure C20. Twelfth Plate, Example Problem.

MINS
16.3



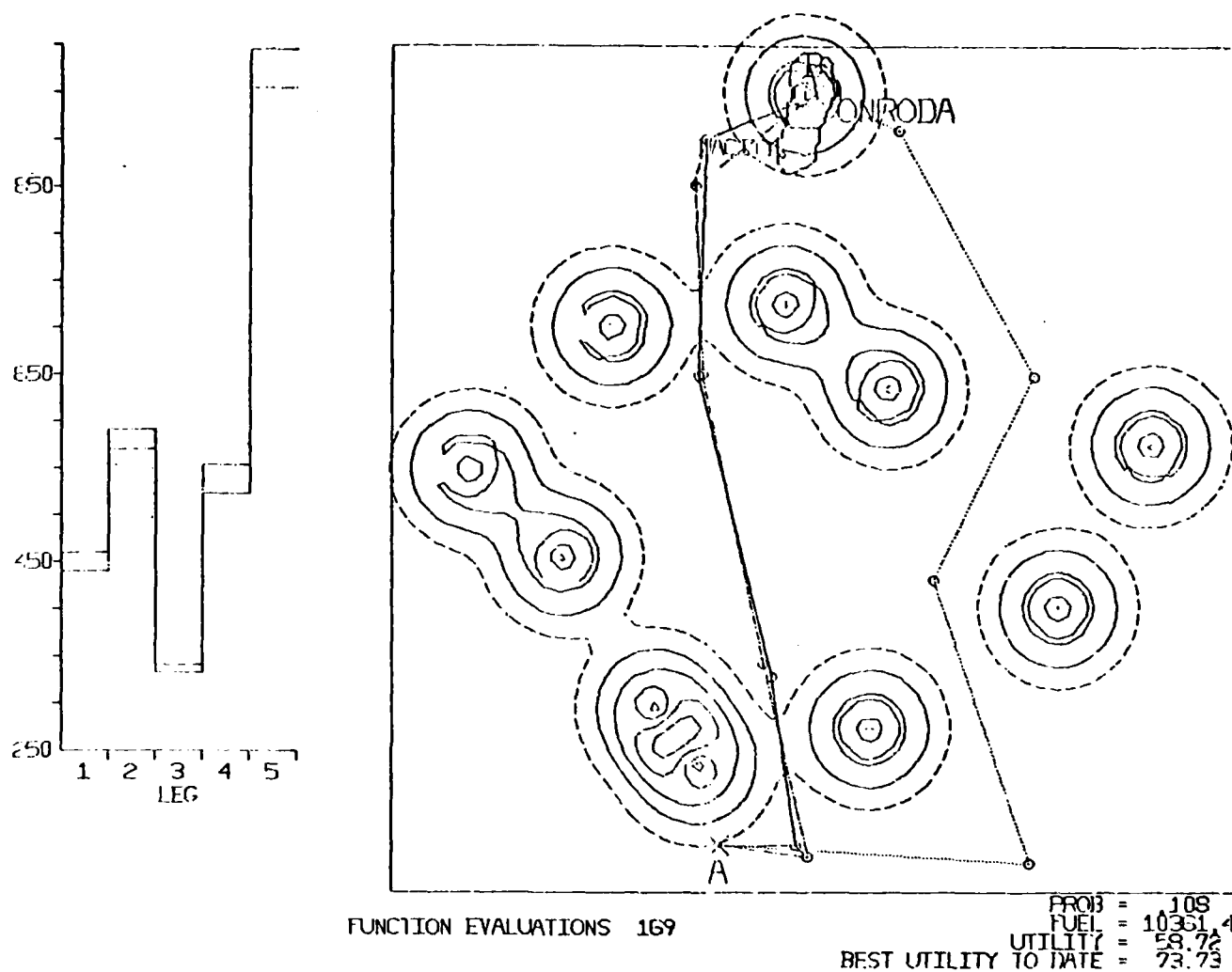
The operator stops the algorithm at the end of 176 evaluations. Note again that the algorithm has moved the last way point much closer to ONRODA. (Disregard time shown under "MINS" from this figure on.)

Figure C21. Thirteenth Plate, Example Problem.



The operator has selected way points for exploring the fourth path and started the algorithm. At the end of 16 evaluations the utility is 56.18.

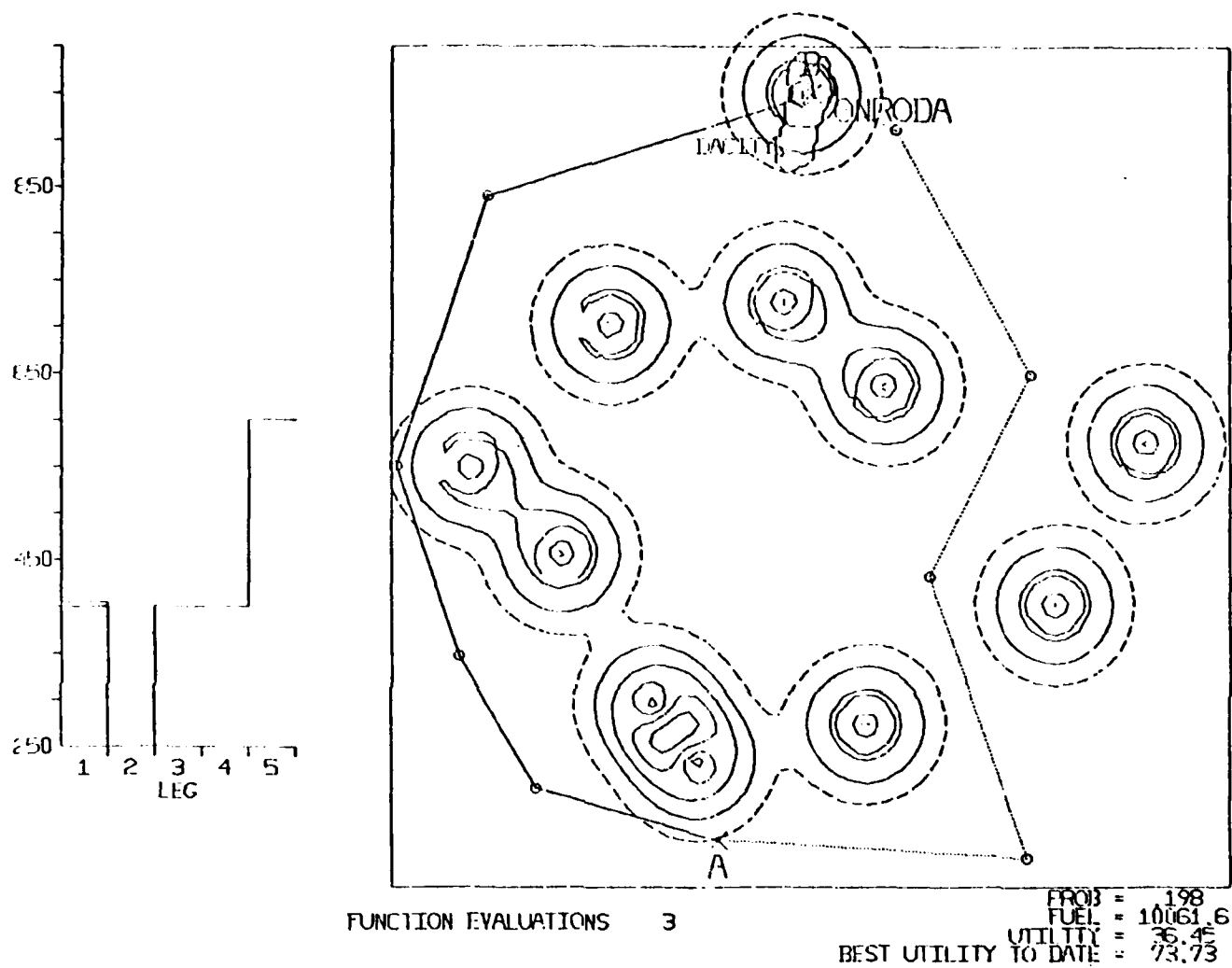
Figure C22. Fourteenth Plate, Example Problem.



The operator stops the algorithm after 169 evaluations. Again, note that the algorithm moved the last way point closer to ONRODA. Utility is competitive with the utility for the first path explored (58.72 versus 56.62) but is significantly lower than the utilities achieved for the second and third paths explored (58.72 versus 73.73 and 67.37).

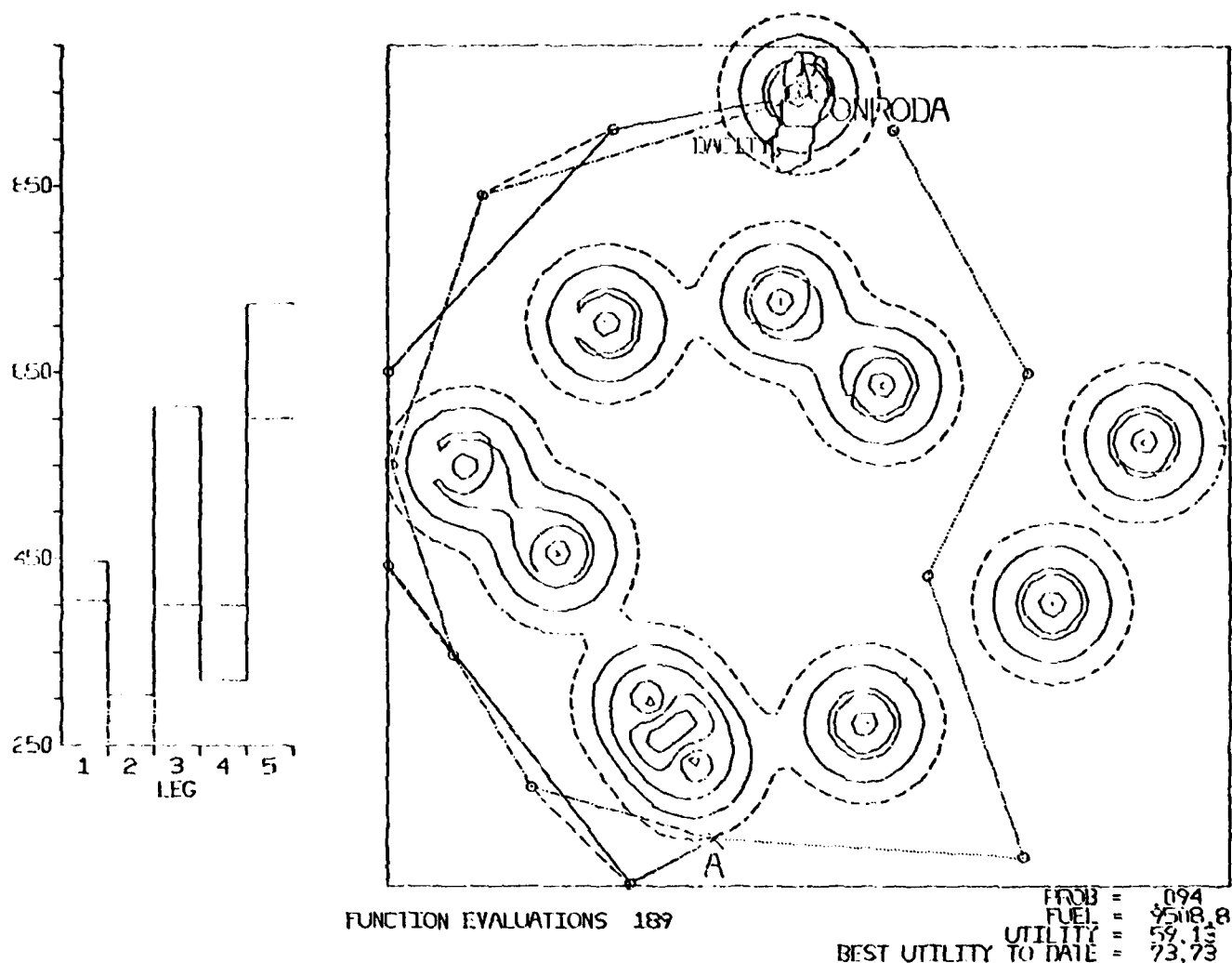
Figure C23. Fifteenth Plate, Example Problem.

MINS
14.1



The operator resets the way points to explore the fifth path. At the end of three evaluations the utility is 36.45.

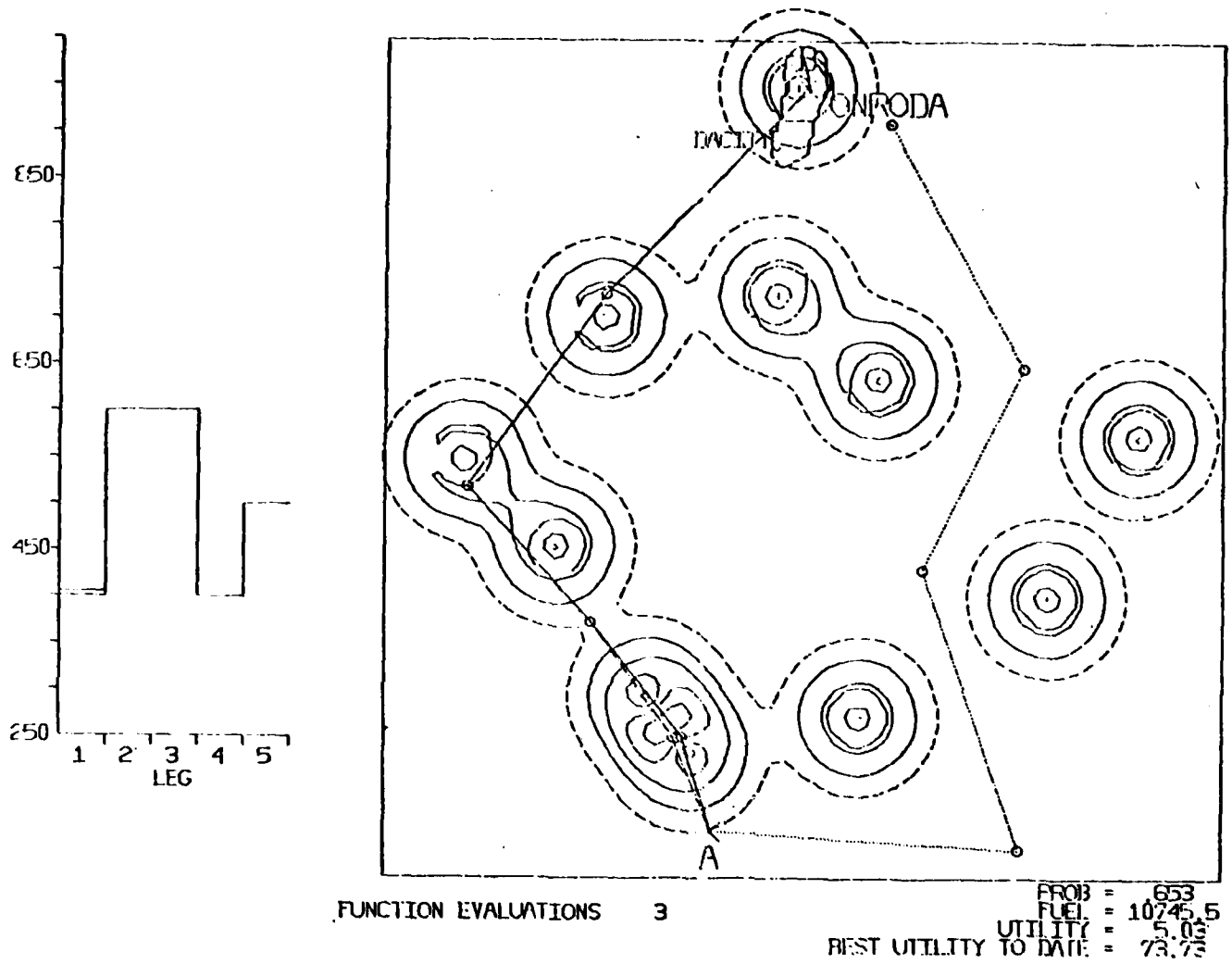
Figure C24. Sixteenth Plate, Example Problem.



The operator stops the algorithm at the end of 189 evaluations. Note that the algorithm moved the last way point closer to ONRODA. Also, note that the first way point was moved down to get away from the contours above the starting point.

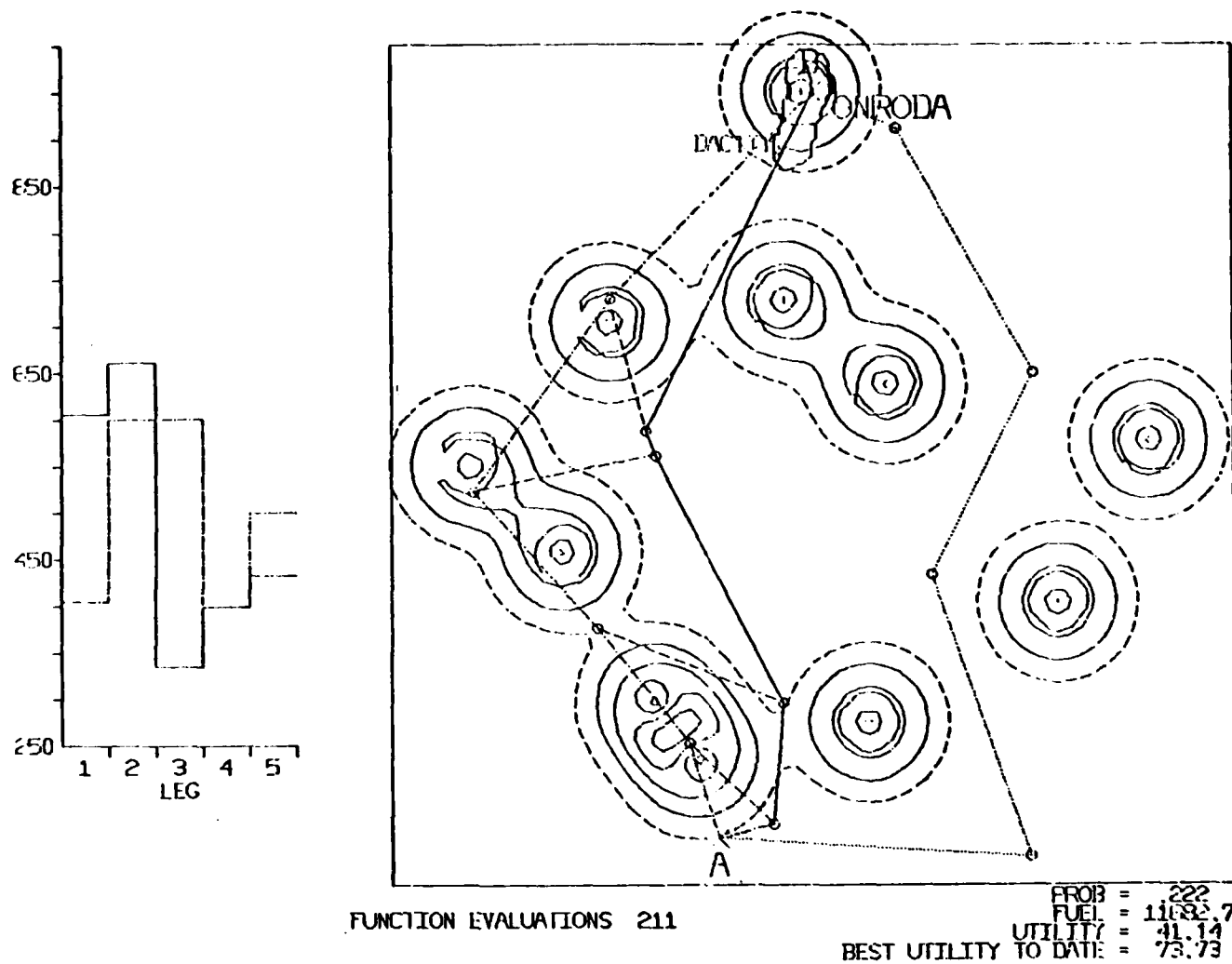
Figure C25. Seventeenth Plate, Example Problem.

MINS
18.2



The operator chooses a very poor set of way points going through high detection capability contours.

Figure 026. Eighteenth Plate, Example Problem.



At the end of 211 evaluations the algorithm found its way over to the vicinity of the fourth path evaluated. But, clearly, it would never have found its way to the best path found by the operator interacting with the algorithm.

Figure C27. Nineteenth Plate, Example Problem.

DISTRIBUTION LIST

Director, Engineering Psychology
Programs (Code 455)
Office of Naval Research
800 North Quincy Street
Arlington VA 22217 (5 cys)

Defense Documentation Center
Cameron Station, Bldg. 5
Alexandria VA 22314 (12 cys)

Dr. Craig Fields
Director, Cybernetics Technology
Office
Defense Advanced Research
Projects Agency
1400 Wilson Boulevard
Arlington VA 22209

Naval Analysis Programs
Code 431
Office of Naval Research
800 North Quincy Street
Arlington VA 22217

Statistics and Probability
Program
Code 436
Office of Naval Research
800 North Quincy Street
Arlington VA 22217

Information Systems Program
Code 437
Office of Naval Research
800 North Quincy Street
Arlington VA 22217

Dr. Jesse Orlansky
Institute for Defense Analysis
400 Army-Navy Drive
Arlington VA 22202

Commanding Officer
ONR Branch Office
ATTN: Dr. Charles Davis
536 South Clark Street
Chicago IL 60605

Commanding Officer
ONR Branch Office
ATTN: Dr. E. Gloye
1030 East Green Street
Pasadena CA 91106

Analysis and Support Division
Code 230
Office of Naval Research
800 North Quincy Street
Arlington VA 22217

Operations Research Program
Code 434
Office of Naval Research
800 North Quincy Street
Arlington VA 22217

Commanding Officer
ONR Branch Office
ATTN: Dr. J. Lester
Bldg. 114, Section D
666 Summer Street
Boston MA 02210

Dr. James McGrath
Navy Personnel Research and
Development Center
San Diego CA 92152

Dr. Donald A. Topmiller
Chief, Systems Effectiveness
Branch
Human Engineering Division
Wright Patterson AFB OH 45433

Dr. Julie Hopson
Human Factors Engineering
Division
Code 604
Naval Air Development Center
Warminster PA 18974

Dr. S. D. Epstein
Analytics
2500 Maryland Road
Willow Grove PA 19090

Dr. G. Hurst
University of Pennsylvania
Wharton School
Philadelphia PA 19174

Dr. Kenneth Gardner
Applied Psychology Unit
Admiralty Marine Technology
Establishment
Teddington, Middlesex TW11 0LN
ENGLAND

Dr. Arthur Siegel
Applied Psychological Services
Science Center
404 E. Lancaster Street
Wayne PA 19087

I.R. Mirman
Assistant for Special Projects
HQ AFSC-DL
Andrews AFB MD 20334

Dr. Rex Brown
Decision Science Consortium
Suite 421
7700 Leesburg Pike
Falls Church VA 22043

Mr. Leslie Innes
Human Factors Wing
Defense & Civil Institute
of Environmental Medicine
P.O. Box 2000
Downsview, Ontario
M3M3BQ
CANADA

Dr. Edgar Johnson
Organizations & Systems Research
Laboratory
U.S. Army Research Laboratory
5001 Eisenhower Avenue
Alexandria VA 22333

Dr. Amos Freedy
Perceptrics, Inc.
6271 Variel Avenue
Woodland Hills CA 91364

Dr. Miley Merkhofer
Stanford Research Institute
Decision Analysis Group
333 Ravenswood Avenue
Menlo Park CA 94025

Dr. C. Kelly
Decisions and Designs, Inc.
8400 Westpark Drive, Suite 600
P.O. Box 907
McLean VA 22101

Mr. Joseph Wohl
MITRE Corporation
Box 208
Bedford MA 01730

Mr. Philip Andrews
Naval Sea Systems Command
NAVSEA 0341
Washington DC 20362

Dr. Robert Kolb
Naval Ocean Systems Center
271 Catalina Boulevard
San Diego CA 92152

Dr. Albert Colella
Combat Control Systems
Naval Underwater Systems Center
Newport RI 02840

Dr. John Silva
Head, Human Factors Division
Naval Ocean Systems Center
San Diego CA 92152

Dr. H.W. Sinaiko
Smithsonian Institution
801 N. Pitt Street
Alexandria VA 22314

Dr. William Dejka
ACCAT Facility
Naval Ocean Systems Center
San Diego CA 92152

Dr. George Pugh
Decision Science Applications, Inc.
1500 Wilson Boulevard
Arlington VA 22209

Dr. Allen C. Miller III
Applied Decision Analysis, Inc.
3000 Sand Hill Road
Menlo Park CA 94025

M. L. Metersky
Naval Air Development Center
Code 5424
Warminster PA 19874

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey CA 92940

Office of the Chief of Naval Operations
OP987H, R & D Plans Division
Pentagon
Washington DC 20350

DATE
FILMED
-8